



UNIVERSIDAD CARLOS III
ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA DE SISTEMAS AUDIOVISUALES

Trabajo Fin de Grado

DESARROLLO DE UN SISTEMA DE INFORMACIÓN SOBRE CONTENIDO
CINEMATOGRAFICO BASADO EN VoiceXML

Alumna: María Fernández Vallejo

Tutor: Dr. David Griol Barres

Septiembre 2012

Título: Desarrollo de un sistema de información sobre contenido cinematográfico basado en VoiceXML

Autora: María Fernández Vallejo

Tutor: Dr. David Griol Barres

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



Agradecimientos

No estaríamos aquí, en este punto, en estos momentos, de no ser por ellos. Gracias a mis padres, por su dedicación, por estos años en los que nunca han faltado las palabras de aliento y apoyo, tan necesarias en muchos momentos. A mi hermana, por esas dosis extra de confianza, a ti, mucha suerte. No me olvido de los ánimos recibidos por parte del resto de mi familia.

Amigas y amigos que habéis estado ahí, viviendo el día a día, o a 660km. Mención para una gaditana por abrirme un poco más los ojos y pensar un poco más en los demás.

Compañeras y compañeros de clase, en especial aquellos con los que alguna vez (o casi siempre) compartí algún grupo de prácticas. Para los que siempre habéis estado dispuestos a echarme una mano, tanto dentro como fuera de la Universidad. Y sobre todo para los que seguís ahí fuera de las aulas. Espero que sepáis quiénes sois.

A mi tutor, David Griol, por sus ánimos en todo momento y por confiar en mí para este proyecto.

A todos vosotros, gracias



Resumen

La aplicación que se describe en el presente documento tiene como principal objetivo superar las barreras tecnológicas que, a menudo, conlleva el manejo de móviles para la navegación en busca de información. Para ello, proponemos la interacción oral con estos dispositivos mediante los denominados sistemas de diálogo hablado.

Los sistemas de diálogo son programas que interactúan con el usuario en lenguaje natural. Sus aplicaciones más conocidas son tareas de provisión de información y servicios web, normalmente a través de un acceso telefónico. A través de estas aplicaciones se facilita el acceso a aplicaciones informáticas en entornos donde utilizando los interfaces tradicionales no sería posible (por ejemplo, en el automóvil), así como se facilita su utilización por personas con discapacidades motoras o visuales. Los módulos más importantes que componen un sistema de diálogo son: un reconocedor del habla que es capaz de reconocer palabras y convertirlas en texto plano y un sintetizador de voz, que realiza la operación inversa. También cuentan con un analizador lingüístico, un módulo que gestiona el diálogo, un generador de respuestas y acceso a bases de datos y fuentes de información.

Para el Trabajo Fin de Grado se ha desarrollado un sistema de diálogo basado en la tecnología VoiceXML, estándar propuesto por el W3C para el acceso oral a la información en Internet, que permite su utilización vía telefónica o sobre VoIP. El intérprete de este lenguaje viene dado por la plataforma de desarrollo de la aplicación, en nuestro caso, Voxeo Evolution. El desarrollo interno de la aplicación se ha programado utilizando además el



lenguaje PHP. Los archivos y ficheros que forman parte del sistema se encuentran alojados en un servidor web externo.

En concreto, el sistema desarrollado aporta información sobre contenido cinematográfico organizada en las siguientes categorías:

- ▶ Información sobre películas: resúmenes, director, actores, premios.
- ▶ Cartelera: el usuario recibe la lista de estrenos en ese momento están estrenadas en España, pudiendo acceder a información sobre ellas.
- ▶ Películas ganadoras de festivales: se puede saber qué películas han ganado las últimas ediciones de premios como los Oscar, los Goya o festivales como el de Venecia.



Abstract

The main objective of the application described in this document is to overcome the technological barriers that often involve managing mobile browsing for information. To this end, we propose oral interaction with these devices via the so-called Spoken Dialogue Systems.

Dialogue systems are programs which interact with the user in natural language. They are known by applications which provide information and web services, usually by telephone access. Through these applications, computer programs can be accessed in environments where using traditional interfaces this access is not possible (e.g., in a car) and facilitates their use by people with visual or motor disabilities. The most important modules that make up a dialogue system are: a speech recognizer that can recognize words and convert them into plain text and a speech synthesizer, which performs the reverse operation. They also have a linguistic analyzer, a module that manages the dialogue, an answer generator and access to databases and information sources.

For this Bachelor Project, a dialogue system has been developed based on VoiceXML technology, which is the standard proposed by the W3C Speech Interface Framework to access oral information. This technology allows users to access the information using his telephone or via VoIP. The interpreter of this language is given by the development application platform, in our case, Voxeo Evolution. The application is also developed by using the PHP programming language. The files required by the system are hosted in an external web server.



In particular, the developed system provides information about movie content organized into the following categories:

- ▶ **Movie Information:** plots, director, actors, awards.
- ▶ **Billboard:** the user receives the list of movies currently being released in Spain; you can then access information about them.
- ▶ **Awarded films festivals:** you can know which movies have won the last editions of the Academy Awards, the Goya Awards or festivals like Venice.



ÍNDICE DE CONTENIDO

| | | |
|-------|--|----|
| 1 | INTRODUCCIÓN | 1 |
| 1.1 | Marco tecnológico..... | 1 |
| 1.2 | Sistemas de diálogo | 3 |
| 1.3 | Motivación..... | 5 |
| 1.4 | Objetivos específicos | 5 |
| 1.5 | Recursos empleados | 7 |
| 2 | ESTADO DEL ARTE..... | 8 |
| 2.1 | ¿Qué es un sistema de diálogo? | 8 |
| 2.1.1 | Sistema de diálogo oral..... | 9 |
| 2.1.2 | Sistema de diálogo multimodal..... | 11 |
| 2.2 | Limitaciones e interacción | 12 |
| 2.2.1 | Tipología de los sistemas de diálogo | 13 |
| 2.3 | El lenguaje VoiceXML (Voice eXtensible Markup Language)..... | 15 |
| 2.3.1 | Conocimientos básico sobre VoiceXML..... | 17 |
| 2.3.2 | Call Control eXtensible Markup Language (CCXML) | 30 |
| 2.3.3 | Speech Recognition Grammar Specification y Semantic Interpretation for Speech Recognition (SRGS) | 30 |
| 2.3.4 | Speech Synthesis Markup Language (SSML) | 30 |
| 3 | ESQUEMA GENERAL DE LA APLICACIÓN | 31 |
| 3.1 | Introducción al sistema | 31 |
| 3.2 | Herramientas | 32 |
| 3.2.1 | Intérprete de VoiceXML: Voxeo Evolution | 32 |
| 3.2.2 | Servidor de alojamiento web: x10hosting..... | 36 |
| 3.2.3 | Obtención de la información: FilmAffinity..... | 39 |



| | | |
|-------|--|----|
| 3.3 | Módulos principales | 40 |
| 3.3.1 | Película | 41 |
| 3.3.2 | Festivales | 42 |
| 3.3.3 | Cartelera | 42 |
| 4 | DISEÑO DE LA APLICACIÓN: MovieBox | 43 |
| 4.1 | Requisitos | 43 |
| 4.2 | Fases de desarrollo | 43 |
| 4.2.1 | Fase previa | 43 |
| 4.2.2 | Fase de desarrollo..... | 45 |
| 4.3 | Módulos | 46 |
| 4.3.1 | Módulo inicial..... | 47 |
| 4.3.2 | Módulo principal..... | 50 |
| 4.3.3 | Módulo auxiliar..... | 58 |
| 4.4 | Evaluación | 65 |
| 5 | PRESUPUESTO Y PLANIFICACIÓN..... | 68 |
| 5.1 | Presupuesto..... | 68 |
| 5.2 | Planificación | 69 |
| 6 | CONCLUSIONES Y TRABAJO FUTURO..... | 71 |
| 6.1 | Conclusiones | 71 |
| 6.2 | Futuras ampliaciones | 74 |
| A. | FAQ (Preguntas frecuentes) | 76 |
| a. | Le he dicho un título de una película y no me ha reconocido, me dice que no me entiende. ¿Qué pasa? | 76 |
| b. | ¿Por qué suena una voz distinta cuando pronuncia un nombre propio en inglés?..... | 77 |
| c. | Al elegir una película mediante un número, ¿por qué no me lleva a la anterior lista? ¿Por qué los números siempre me llevan a la lista nueva?..... | 77 |
| d. | ¿Por qué a veces escucho más tiempo la música y otras veces tan solo el principio? | 77 |
| | Bibliografía | 78 |



ÍNDICE DE FIGURAS

| | |
|--|----|
| Fig. 1 Esquema general de un sistema de dialogo oral | 9 |
| Fig. 2 Esquema general de un sistema de diálogo multimodal | 11 |
| Fig. 3 Diálogo servicio de taxis | 13 |
| Fig. 4 Diálogo guiado | 14 |
| Fig. 5 Arquitectura VoiceXML | 16 |
| Fig. 6 Autómata VXML | 18 |
| Fig. 7 HolaMundo.vxml..... | 18 |
| Fig. 8 Formulario VXML | 24 |
| Fig. 9 Menú VXML | 25 |
| Fig. 10 Detalle <goto> | 26 |
| Fig. 11 Declaración gramática – voz..... | 27 |
| Fig. 12 Declaración gramática – local | 27 |
| Fig. 13 Declaración gramática – externa..... | 27 |
| Fig. 14 sellInfo.xml | 28 |
| Fig. 15 Diagrama aplicación con documento raíz | 29 |
| Fig. 16 Documento raíz y documento hijo | 29 |
| Fig. 17 Esquema cliente–servidor..... | 32 |
| Fig. 18 Creación de cuenta en Voxeo | 33 |
| Fig. 19 Detalle vista general cuenta Voxeo | 34 |
| Fig. 20 Registro aplicación Voxeo | 35 |
| Fig. 21 Información de contacto..... | 36 |
| Fig. 22 Enlace a fichero inicial desde Voxeo..... | 36 |
| Fig. 23 Panel de control x10hosting | 38 |
| Fig. 24 Administrador de ficheros x10hosting | 39 |
| Fig. 25 Esquema general de la aplicación..... | 40 |
| Fig. 26 Diagrama organización aplicación..... | 41 |
| Fig. 27 Prototipo estático de la aplicación..... | 45 |
| Fig. 28 Prototipo con servidor externo | 45 |
| Fig. 29 Prototipo dinámico de la aplicación..... | 46 |
| Fig. 30 Bienvenida al sistema MovieBox..... | 47 |
| Fig. 31 Diálogo petición título..... | 48 |



| | |
|--|----|
| Fig. 32 Detalle <submit> movie_search..... | 49 |
| Fig. 33 Diálogo petición cartelera..... | 49 |
| Fig. 34 Diálogo película y valoración | 49 |
| Fig. 35 Detalle <submit> billboard_search.php..... | 50 |
| Fig. 36 Diálogo petición nombre del festival | 50 |
| Fig. 37 Detalle <submit> prize_search.php | 50 |
| Fig. 38 Utilización de PHP conjuntamente con VoiceXML | 51 |
| Fig. 39 Flujo información general movie_search..... | 51 |
| Fig. 40 Resultado de búsqueda de título: un único resultado | 52 |
| Fig. 41 Resultado de búsqueda de título: más de un resultado | 53 |
| Fig. 42 Diagrama de bloques movie_search | 54 |
| Fig. 43 Diálogo Cartelera General (15 Agosto 2012)..... | 55 |
| Fig. 44 Cartelera Puntuación (15 Agosto 2012)..... | 55 |
| Fig. 45 Diagrama de bloques billboard_search..... | 57 |
| Fig. 46 Llamada método de búsqueda | 58 |
| Fig. 47 Sinopsis resultado método de búsqueda | 59 |
| Fig. 48 Fichero selltem.xml generado para consulta 'Gladiator' | 60 |
| Fig. 49 Diálogo resultado múltiple de la búsqueda de título | 61 |
| Fig. 50 Caracteres reemplazables..... | 61 |
| Fig. 51 \$SP: sinopsis en español | 63 |
| Fig. 52 \$EN: sinopsis en inglés..... | 63 |
| Fig. 53 Sinopsis bilingüe..... | 63 |
| Fig. 54 link.php..... | 64 |
| Fig. 55 Diagrama de Gantt..... | 70 |
| Fig. 56 Dialogo preciso..... | 71 |
| Fig. 57 Dialogo redundante | 72 |



ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1 Elementos de la comunicación | 3 |
| Tabla 2 Ventajas e inconvenientes del uso de base de datos..... | 6 |
| Tabla 3 Elemento VXML –Atributo – Función | 21 |
| Tabla 4 Atributo – Descripción..... | 22 |
| Tabla 5 Eventos predefinidos en VoiceXML | 28 |
| Tabla 6 Número de identificación de película..... | 59 |
| Tabla 7 Valoración conocimientos previos | 65 |
| Tabla 8 Valoración uso previo de interfaces orales | 65 |
| Tabla 9 Valoración comprensión de los mensajes | 65 |
| Tabla 10 Valoración velocidad de interacción..... | 66 |
| Tabla 11 Valoración dificultad..... | 66 |
| Tabla 12 Valoración obtención de la información | 66 |
| Tabla 13 Valoración satisfacción de uso de la aplicación | 66 |
| Tabla 14 Valoración seguimiento del diálogo..... | 67 |
| Tabla 15 Valoración comportamiento similar a ser humano | 67 |
| Tabla 16 Valoración global de la aplicación | 67 |
| Tabla 17 Planificación..... | 69 |



Capítulo 1

1 INTRODUCCIÓN

La función de esta sección introductoria es situar al lector dentro del marco tecnológico que ha dado lugar al desarrollo del presente proyecto. Además, se describe de manera genérica cuáles han sido los objetivos y el porqué de este trabajo.

En los siguientes capítulos y apartados se detallará la evolución de esta tecnología y los pormenores en el desarrollo de la aplicación que nos ocupa.

1.1 Marco tecnológico

Vivimos en un mundo basado en la tecnología y ésta, además, en continua evolución. Casi en cuestión de meses nuestras últimas adquisiciones, sobre todo en materia de telefonía e informática, quedan obsoletas por el lanzamiento de una última versión del producto.

Todo este avance se fundamenta en hacer más fácil la vida del ser humano. Ahora, gracias a las tecnologías inalámbricas y a los modernos teléfonos móviles tenemos acceso a internet donde quiera que nos encontremos. Podemos acceder a cualquier información disponible en la red y en cuestión de segundos se nos presenta en pequeñas pantallas de no más de 4 pulgadas. Estos dispositivos carecen de teclado, a menos que se le conecte uno externo, y toda interacción con ellos se realiza a través de estas pantallas táctiles.

Sin embargo, no todo son ventajas en cuanto al avance de estos terminales. Situémonos en la navegación por internet que se realiza en un ordenador. A golpe de ratón somos capaces de abrir tantas pestañas en nuestro navegador como queramos y saltar de una a otra con la misma facilidad. En estos mini ordenadores, a veces para poder enlazar con una nueva página hay que aumentar el zoom de la vista actual y poder seleccionar cómodamente el nuevo enlace que puede o no abrirse en una nueva página o sobrescribir la actual. Al final, disponemos de una toda una presentación de páginas a las que sólo es posible acceder a la última si se ha accedido previamente a todas las anteriores, por citar un ejemplo. A simple vista, no parece un gran inconveniente, pero no todas las personas disponen de esa destreza dactilar que parece imprescindible para manejar dichos aparatos.

Los últimos modelos de teléfonos móviles incluyen como novedad herramientas conocidas como reconocedores automáticos del habla (Automatic - Speech - Recognition ASR). Básicamente, la idea es la siguiente: el usuario dice una frase más o menos sencilla y el sistema es capaz de plasmar dicha frase en una secuencia de palabras. Su utilización en móviles está pensada para “escribir” mensajes de texto sin necesidad de pulsar las teclas, o notas tipo memorándum, etc. No obstante, esta tecnología no es nueva, si no que de forma paralela y sin tanta publicidad, se ha integrado desde hace años en los denominados sistemas de diálogo [PIE12] [LOP05] [McT04] para la comunicación hombre-máquina. Y es que, somos seres comunicadores por naturaleza y, en la mayoría de las ocasiones, la base para esta comunicación es la

voz. Por tanto, no es de extrañar que cada vez más dispositivos incluyan esta característica entre sus múltiples funcionalidades.

1.2 Sistemas de diálogo

El objeto de este tipo de programas es facilitar cierta información o servicios entablando una conversación hombre-máquina. Los sistemas de diálogo se encuentran adaptados a personas con discapacidad, ya que existen diferentes modelos de interacción con ellos que no se restringen exclusivamente al uso del habla, también pueden utilizarse otros dispositivos manuales. Según la interfaz de entrada y salida se pueden clasificar en dos tipos:

- **Sistema hablado**

Utilizan la voz como vía de entrada y salida de la información. Para llevar a cabo una satisfactoria interacción hombre - máquina se consideran todos los elementos existentes en la comunicación. La Tabla 1 muestra los principales bloques que forman parte de este tipo de sistemas. En la próxima sección se detallan los módulos ASR y TTS.

| | |
|-----------------------|--|
| Emisor humano | Reconocedor automático del habla (ASR) |
| Emisor máquina | Sintetizador de voz (TTS) |
| Canal | Cable telefónico, vía web (VoIP) |
| Código | Idioma de la aplicación |
| Contexto | Diseño de la aplicación |

**Tabla 1 Elementos de la comunicación
en los sistemas de diálogo**

- **Sistema multimodal**

En este tipo de sistemas cualquier dispositivo de entrada es válido. Igualmente, la salida de respuestas también puede generarse a través de texto, imágenes, etc.

El presente proyecto está basado en los sistemas de diálogo hablado, concretamente los desarrollados mediante el estándar VoiceXML [VOI]. Se trata de un lenguaje basado en etiquetas XML. Las aplicaciones creadas con VXML se asemejan bastante, en lo que a interpretación se refiere, a las programadas con HTML. Los códigos son leídos por cada intérprete y son capaces de generar y presentar un resultado dependiendo de las acciones de los usuarios.

Dada la versatilidad de este estándar se puede desarrollar una amplia gama de aplicaciones. El sistema desarrollado está enfocado en dar información sobre contenidos de películas tales como su sinopsis, cuántos premios tiene o quien la ha dirigido. Además, proporciona la cartelera para la semana actual, pudiendo acceder nuevamente a información más específica sobre las películas en cartel. Una de las ventajas principales que aporta este sistema es la rapidez en el acceso a la información. No es necesario esperar a que el dispositivo acceda a internet, después utilizar el buscador para a su vez buscar la película, etc. Simplemente con una llamada de teléfono dispondremos de esta información. Se trata de un acceso tan sencillo como podría ser ir por la calle, ver el título de una película, llamar y saber de qué va, quién la ha dirigido o qué actores la protagonizan.

Más adelante se detalla una explicación tanto del lenguaje VoiceXML como de las herramientas para su desarrollo y codificación. Igualmente se hará un detenido estudio de cada uno de los bloques que conforman la aplicación así como su interacción entre ellos.

1.3 Motivación

Con la realización de este Trabajo Fin de Grado se marcaron unos objetivos relativos tanto a la aplicación, utilidad, robustez, etc., así como relativas al desarrollo de habilidades y conocimiento personales.

Para la aplicación se buscaba básicamente utilidad y que fuera manejable para cualquier persona. Aunque se trata de una aplicación para dar información de contenido cinematográfico, no se restringe únicamente a la comunidad de cinéfilos, ya que la información aportada es de carácter generalista. Está pensada para todas aquellas personas que ven el título de una película y les gustaría saber de qué va, o por ejemplo, en la propia cola del cine, donde puede primar la rapidez, saber de qué va o cuántos premios tiene la película cuya sesión empieza en menos de un cuarto de hora. Otra posible situación es querer conocer la cartelera de la semana mientras estamos andando por la calle, elegir una película, interesarte por ella e ir al cine.

En cuanto a los objetivos de aprendizaje y desarrollo, rara vez se sabe cuál va a ser el resultado. No sabemos a priori qué vamos a aprender o qué habilidades se van a desarrollar. Al empezar este proyecto desconocía totalmente tanto el lenguaje VXML como PHP, indirectamente se habían convertido en los principales propósitos si quería construir una buena aplicación. La más o menos rápida adquisición de estos conocimientos han demostrado la capacidad de adaptación a nuevos entornos, además de resultar útiles.

1.4 Objetivos específicos

En este caso, la prioridad específica de la aplicación ha sido el funcionamiento dinámico de la misma. Como se ha explicado en anteriores secciones, la información para la generación de las respuestas puede obtenerse mediante consultas a base de datos, o bien accediendo a otro tipo de ficheros para recuperar la información. Ambos sistemas tiene sus pros y sus contras, tal y como muestra la Tabla 2.

| | Ventajas | Inconvenientes |
|----------------|----------|--|
| Bases de datos | Rapidez | Actualización con cada nueva información |
| Acceso a webs | Dinámica | Acceso a la red variable |

Tabla 2 Ventajas e inconvenientes del uso de base de datos

Esta aplicación en concreto carece de base de datos. Toda la información se recoge de la web FilmAffinity (<http://www.filmaffinity.com>). Por tanto, su funcionamiento dinámico está garantizado.

A continuación se citan otros objetivos fundamentales definidos para la aplicación:

- **Precisión:** a la hora de responder a las cuestiones del usuario no hacen falta más adornos que decir directamente la respuesta. Si se quiere conocer el nombre del director de la película es lo que debe responderse. Respuestas concretas para preguntas concretas.
- **Brevedad en los discursos del TTS:** personalmente, encuentro bastante tedioso los sintetizadores de voz cuyos mensajes informativos se eternizan. Esto lleva a la desesperación del usuario y por tanto a dejar de usar la aplicación. Uno de los objetivos para el desarrollo de este proyecto es evitar que esta situación ocurra, limitando en la medida de lo posible las instrucciones dadas por el TTS.
- **Instrucciones sencillas:** para facilitar el uso de la aplicación las instrucciones han de ser sencillas, sin gramáticas complejas que hagan que el usuario esté repitiendo constantemente la solicitud de datos al sistema.

- **Claridad en las expresiones del TTS:** si bien hay enunciados que escapan a nuestro control, ya que se recuperan directamente de la web, se puede hacer lo posible para hacer estos más comprensibles. En concreto, el uso de nombres propios extranjeros y su lectura “españolizada” podrían ser un problema.
- **Número limitado de opciones:** no en detrimento del usuario, sino para evitar tener que memorizar una lista tan amplia de posibles opciones. Seguramente cuando se haya alcanzado el final de la lista, la primera se haya esfumado de la cabeza.

1.5 Recursos empleados

Las herramientas utilizadas para el desarrollo de esta aplicación han sido:

Hardware

- ▶ Ordenador portátil con micrófono integrado.
- ▶ Servidor de VoiceXML: Voxeo.
- ▶ Servidor web con soporte para PHP: x10hosting.

Software

- ▶ Plataforma de desarrollo Voxeo Evolution [VOX]
- ▶ Navegador web: Google (<http://www.google.es/>)
- ▶ Editor de texto: Notepad++



Capítulo 2

2 ESTADO DEL ARTE

Antes de profundizar más en el desarrollo e implementación de la aplicación, conviene alejar un poco el foco y explicar más en detalle qué es un sistema de dialogo, en qué consiste, qué otros estándares existen.

2.1 ¿Qué es un sistema de diálogo?

Como ya se introdujo en el primer capítulo, los sistemas de diálogo o sistemas conversacionales (Spoken Dialog Systems) se desarrollaron para interactuar con los usuarios de manera oral, en su mayoría, aunque también existen los llamados multimodales. Son programas informáticos capaces de simular un diálogo como si de una persona se tratase. Su función es reconocer qué es lo que se les está diciendo, analizarlo y producir una respuesta coherente y lógica. Suelen utilizarse para tareas más o menos automáticas como proporcionar información, gestionar venta de billetes de transporte, un pedido a domicilio, etc.

2.1.1 Sistema de diálogo oral

La Figura 1 resume la arquitectura típicamente utilizada para el desarrollo de estos sistemas.

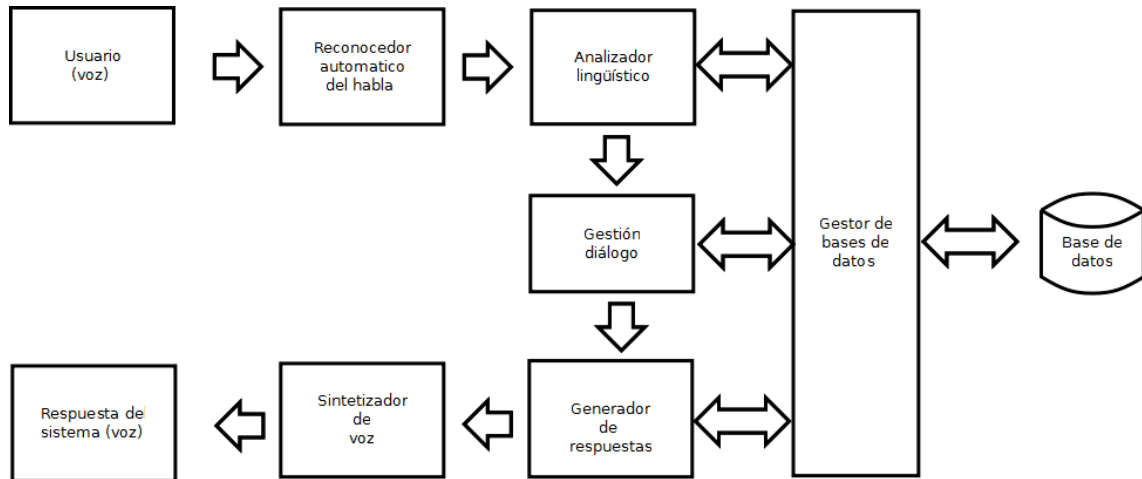


Fig. 1 Esquema general de un sistema de dialogo oral

- **Reconocedor automático del habla (ASR – Automatic Speech Recognition)**

Entrada al sistema. Este módulo se encarga de procesar la onda sonora para obtener la secuencia de palabras mencionadas por el usuario. Tras el estudio estadístico de la señal, normalmente mediante Modelos Ocultos de Markov [MAR], devuelve aquellas palabras más probables de haber sido pronunciadas.

Las palabras que se reconocen no son meras sílabas unidas al azar que dan como resultado algo coherente. Estas forman parte de un diccionario o gramática, como veremos más adelante, que recoge las palabras que el sistema será capaz de reconocer. Unas de las mayores dificultades a la hora de diseñar estos sistemas, radica en la imposibilidad de predecir qué va a decir el usuario, si utilizará o no abreviaturas, frases gramáticamente incompletas, etc. De ahí, la importancia de hablar de forma clara y sin titubeos que

puedan confundir al sistema. Además, cubrir todos los potenciales casos para un dominio determinado es una tarea prácticamente imposible.

- **Analizador lingüístico:**

Hasta ahora las frases eran una sucesión de palabras elegidas por probabilidad de una lista. En este módulo se les aporta significado para la aplicación mediante la utilización de gramáticas o métodos estadísticos. [GLA95] [HE03]

- **Gestión de diálogo:**

Se encarga de manejar la conversación dentro del dominio del sistema. Está al corriente de la situación actual del diálogo y en función de lo que el usuario ha dicho y la información que pueda precisar de la base de datos o fuente de información impulsa la conversación hacia una determinada acción u otra. [GRI08][HUR05]

- **Generador de respuestas:**

Como su propio nombre indica, este módulo es el encargado de formar la respuesta que proporcionar al usuario. Por “formar respuesta” se entiende construir una frase en lenguaje natural, coherente, con sentido y conforme a la consulta del usuario.

- **Sintetizador de voz (Text To Speech TTS):**

Este bloque es capaz de conferir voz al sistema mediante la conversión a señal de voz de la respuesta generada por el bloque anterior, es decir, no grabada previamente.

- **Gestor de bases de datos:**

Se encarga del acceso a la base de datos. Las consultas las realizan los módulos centrales, éste las gestiona y obtiene una respuesta

- **Base de datos:**

Almacén de datos que puede solicitar la aplicación. No es imprescindible, ya que hay otras formas de obtener información (por ejemplo, directamente de páginas web).

2.1.2 Sistema de diálogo multimodal

A diferencia del sistema oral, donde como entrada únicamente se acepta la voz, los sistemas multimodales son menos estrictos, se puede acceder a ellos utilizando cualquier interfaz de entrada como tipo texto, imagen, etc. [DON10].

Básicamente, estos sistemas contienen los mismos módulos que los sistemas orales, puesto que también aceptan voz, incorporando adicionalmente gestores de entrada y salida para las distintas modalidades que contempla el sistema (Figura 2).

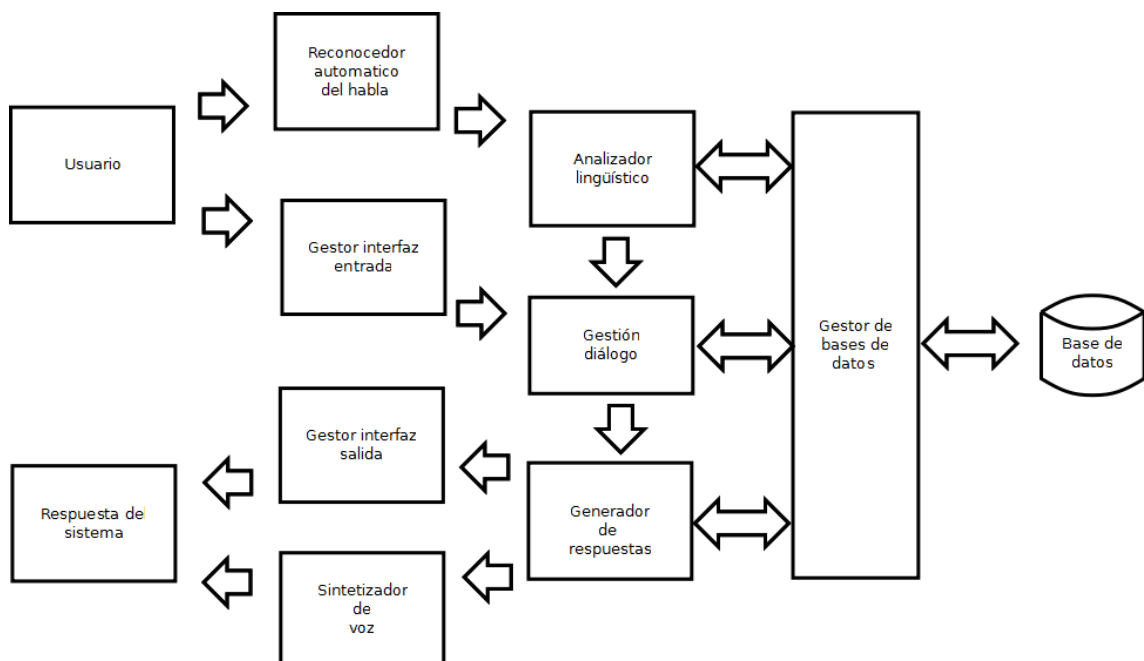


Fig. 2 Esquema general de un sistema de diálogo multimodal

2.2 Limitaciones e interacción

Si bien se ha comentado anteriormente que los sistemas de diálogo hablado están capacitados para mantener una conversación, en realidad no es del todo cierto, existen limitaciones. Éstas afectan, en cierta medida, al modo en el que nos comunicamos con ellos. No olvidemos que se trata de máquinas, no de seres humanos [UGR].

Las aplicaciones que se desarrollan en base a un sistema de diálogo están condicionadas por la función que desempeñan. Son aplicaciones, más o menos especializadas, que están entrenadas para recibir enunciados de contenido concreto. Esto se refleja en la capacidad de síntesis de estos programas. Dentro de un dominio específico, reconocerán únicamente palabras clave para desempeñar su función. Aunque cada vez más cerca, todavía no se ha desarrollado el módulo de reconocimiento del habla que sea capaz de descifrar y interpretar cualquier mensaje en cualquier circunstancia.

Supongamos por ejemplo que una persona llama a una aplicación de servicios de taxis: *“Hola, buenas noches, quiero pedir un taxi. Estoy en la calle Pez Espada, debajo de la tercera farola, junto al número 4”*. Del enunciado anterior, el sistema de taxis únicamente reconocerá dónde se encuentra el usuario; ni siquiera qué es lo que precisa, puesto que se ha pedido un servicio a un sistema de taxis. Además, según se observa en la Figura 3, aunque en una primera intervención el usuario ya indica en qué número se encuentra, el sistema en ese momento solo precisaba el nombre de la calle, por tanto, es lo único que reconocerá. Si se tratara de un sistema ideal, éste comprendería la frase completa, y respondería con algo parecido a *“De acuerdo, en seguida le mandamos uno”*. Sin embargo, dadas las restricciones en el reconocimiento, nuestro sistema debe confirmar que ha recibido correctamente la información.

| | |
|----------|---|
| Sistema: | <i>Bienvenido al servicio de taxis. Diga en qué calle se encuentra.</i> |
| Usuario: | <i>Hola, buenas noches, quiero pedir un taxi. Estoy en la calle Pez Espada, debajo de la tercera farola, junto al número 4.</i> |
| Sistema: | <i>Ha dicho que la calle se llama Pez Espada. ¿Es correcto?</i> |

| | |
|----------|--|
| Usuario: | <i>Sí.</i> |
| Sistema: | <i>Por favor, diga el número.</i> |
| Usuario: | <i>4</i> |
| Sistema: | <i>Ha dicho que el número es el 4. ¿Es correcto?</i> |
| Usuario: | <i>Sí.</i> |
| Sistema: | <i>En seguida llegará uno. Gracias por utilizar el servicio.</i> |

Fig. 3 Diálogo servicio de taxis

De la Figura 3 podemos obtener una idea preliminar del formato de interacción de estos sistemas. Grosso modo, se basan en formularios cuyos campos se rellenan para componer una lista de datos, procesarla y elaborar una respuesta.

2.2.1 Tipología de los sistemas de diálogo

A continuación se describe una posible catalogación de los sistemas de diálogo. No todos los criterios son excluyentes, es decir, un sistema puede ser guiado multimodal, pero no podrá ser monolingüe y políglota al mismo tiempo.

Sistemas de diálogo guiados

En este tipo de sistemas el diálogo transcurre con un continuo intercambio de mensajes entre la aplicación y el usuario.

La conversación es iniciada normalmente por la máquina, no obstante, es el usuario quien primero realizó la llamada con el fin de obtener una información. Éste se limita a responder a las preguntas que le va planteando el sistema utilizando para ellos respuestas simples especificadas por la aplicación tales como sí o no, marcación de teclas en el teléfono o alguna palabra indicada para ello.

Este tipo de diálogos son muy limitados para el usuario, ya que las respuestas son acotadas, además de no aceptar interrupciones (Figura 4). Sin embargo, desde el punto de vista de diseño son los más sencillos, ya que es el programador quien marca la línea que seguirá la conversación, adaptando a ella las posibles respuestas.

| | |
|----------|---|
| Sistema: | <i>Bienvenido a Shopping List, el sistema para hacer la lista de la compra. Por favor siga las instrucciones, diga qué.</i> |
| Usuario: | <i>Pero yo....</i> |
| Sistema: | <i>producto necesita: carne, verdura, pescado o fruta</i> |
| Usuario: | <i>Cereales.</i> |
| Sistema: | <i>Diga qué producto necesita: carne, verdura, pescado o fruta.</i> |
| Usuario: | <i>Fruta.</i> |
| Sistema: | <i>Ha dicho fruta. Diga qué fruta necesita: plátano, manzana, naranja, fresa. [...]</i> |

Fig. 4 Diálogo guiado

Sistemas de diálogo cooperativo o mixto

Son menos restrictivos que los anteriores, permitiendo al usuario interrumpir el discurso de la aplicación para adelantarse a la explicación o bien indicar otra acción. Por parte de la aplicación se incluyen diálogos para avisar de entradas erróneas introducidas por el usuario.

Sistemas de diálogo adaptativo

Son los más complejos de diseñar. El sistema es “inteligente” y va aprendiendo nuevas pautas de comunicación con el usuario conforme los nuevos diálogos que vaya completando.

Este tipo de sistemas puede incluso incorporar un análisis de las emociones que está experimentando el usuario, pudiendo así tratarlas adecuadamente (en especial, emociones negativas como el enfado, la duda o el aburrimiento) e incluso anticiparse a ellas. Sin embargo, el registro de estas emociones es lo que los hace tan complejos. Han de existir bases de datos de personas manteniendo una conversación en un estado de ánimo concreto, pero se necesita un conjunto lo bastante amplio como para ser capaz de reconocer alegría o enfado en cualquier persona a través de la voz. El diálogo fluye de manera más parecida a cómo se haría con un ser humano.

Sistemas de diálogo monolingües y multilingües

Las aplicaciones desarrolladas monolingües “hablan y reconocen” un solo idioma. Los sistemas multilingües dan la opción al usuario de elegir el idioma para el diálogo. Podría darse el caso de utilizar únicamente otro idioma para casos concretos como decir el nombre de ciudades o de personas.

Sistemas de diálogo unimodales y multimodales

Los sistemas unimodales son sistemas que sólo aceptan como entrada la voz. En cambio, tal y como se ha descrito previamente, los sistemas de diálogo multimodales pueden accederse y generar diferentes modalidades aparte de la voz, como la marcación de teclas en el teléfono (DTMF) por ejemplo. Se trata de sistemas que cuentan con gestores de interfaces de entrada aparte del módulo de reconocimiento del habla.

2.3 El lenguaje VoiceXML (Voice eXtensible Markup Language)

VoiceXML es un lenguaje de marcado, basado en XML. La primera versión (VoiceXML 1.0) se presentó por primera vez en Marzo de 2000 y posteriormente, VoiceXML 2.0 ha sido recomendada por la W3C (World Wide Web Consortium) como uno de los estándares para el desarrollo de aplicaciones de sistemas de diálogo. Para llevar a cabo una fluida comunicación hombre máquina, a parte de la implementación de los bloques básicos como ASR o TSS y el manejo de datos, VoiceXML dispone además de control del flujo del diálogo, permite la salida de audio grabado así como la grabación de un mensaje por el canal de entrada, reconocimiento de tonos (DTMF).

La gestión de la comunicación sigue el esquema mostrado en la Figura 5. Los diferentes bloques se describen a continuación.

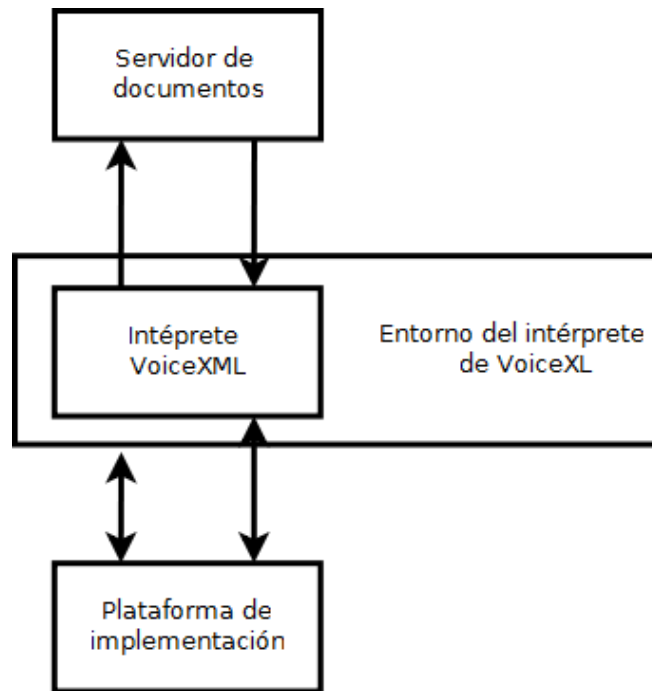


Fig. 5 Arquitectura VoiceXML

- **Intérprete de VoiceXML**

El intérprete del lenguaje es quien se encarga de seguir el guión programado por el desarrollador de la aplicación. Éste hará uso del servidor de documentos para generar aquellos que sean solicitados para dar respuestas al usuario. Además de aportar información obtenida por otros documentos, también es capaz de acceder a la web y recuperar otras informaciones dinámicas.

- **Servidor de documentos**

Se encarga de la gestión de los ficheros que forman la aplicación. Existen varias posibilidades de ubicación:

- Fuera de la plataforma de implementación: de forma que desde la plataforma se indique la dirección del alojamiento del fichero raíz.
- Dentro de la plataforma: los ficheros se encuentran contenidos dentro de la propia plataforma de desarrollo. Hay que tener cuidado con el tipo de ficheros, ya que no

todas las plataformas de implementación soportan todo tipo de archivos (.PHP, .JS, etc.).

- Mixta: es decir, existen ficheros dentro de la plataforma y otros que están fuera. Normalmente el fichero raíz es el que se incluye en la plataforma y es capaz de comunicarse con el resto alojados en uno externo.

- **Entorno del intérprete de VoiceXML**

La llamada del usuario para iniciar la comunicación es respondida por el entorno del intérprete. Además es capaz de realizar otras tareas de forma independiente y paralela como hacer un seguimiento de las entradas del usuario.

- **Plataforma de implementación**

Depende tanto del intérprete como de su entorno. Su funciones se basan de dar respuesta a algunos eventos generador por el usuario, como la pulsación de teclas del teléfono, o a lo generados por el propio sistema como la expiración temporizadores

2.3.1 Conocimientos básico sobre VoiceXML

En este apartado completamos un recorrido por los conceptos más importantes del estándar VoiceXML [VXML], con el principal objetivo de ayudar al lector a entender un poco más sobre los sistemas de diálogo y su programación interna, así como comprender el funcionamiento de la aplicación desarrollada para el Trabajo Fin de Grado.

Entiéndase una aplicación VXML como una máquina de estados. Según la actuación del usuario, sus elecciones previas, el fin de su consultase irán sucediendo los diálogos y documentos hasta completar el ciclo y se genere la respuesta al usuario. Los diálogos son una analogía de los estados del autómeta (Figura 6)

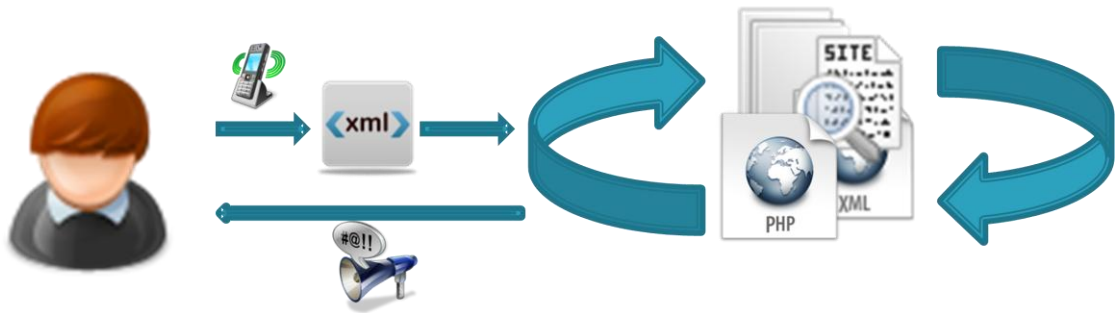


Fig. 6 Autómata VXML

Un enfoque ascendente (Down-top approach)

Al tratarse de un lenguaje de marcado, está compuesto por un conjunto de etiquetas y el programa completo es interpretado por un servidor de voz. Estas etiquetas van enmarcadas dentro de los símbolos “< >” e identifican el tipo de elemento del que se trata. En la Figura 7, se muestran varios ejemplos de elementos incluidos dentro del documento HolaMundo.vxml¹.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <vxml version = "2.1" >
3
4      <form>
5          <block>
6              <prompt>
7                  Hola Mundo. Esta es mi primera aplicación de VXML.
8              </prompt>
9          </block>
10     </form>
11 </vxml>

```

Fig. 7 HolaMundo.vxml

Observaciones:

- Todas las líneas, excepto la 3 y la 7 son etiquetas.
- A diferencia del resto, los elementos de las líneas 1 y 2 además tienen propiedades, son los llamados atributos. Se observa la diferencia en que después del “<” y el nombre del elemento (<form>) hay más texto (<vxml version = "2.1" >).

¹ HolaMundo.ext es por antonomasia primer programa que se implementa para introducir el estudio de un nuevo lenguaje de programación. Se diseña por su sencillez, ya que únicamente muestra por pantalla el texto Hola Mundo.

La Tabla 3 muestra una lista con los principales elementos definidos en el estándar VoiceXML.

| Elemento | Atributo | Descripción |
|--------------|--|--|
| <assign> | expr, name | Asigna el valor <i>expr</i> a una variable <i>name</i> . |
| <audio> | expr, src, fetchhint, fetchtimeout | Reproduce audio <i>expr</i> |
| <block> | cond, expr, name | Contiene código ejecutable, no interactivo |
| <catch> | cond, count, event | Captura el evento <i>event</i> |
| <choice> | accept, dtmf, event, next, eventexpr, maxstale, expr, fetchaudio, fetchhint, fetchtimeout, maxage, message | Define un elemento de un menú (ítem) |
| <clear> | Namelist | Limpia una o varias variables contenidas en <i>namelist</i> de un formulario |
| <disconnect> | Namelist | Desconecta la sesión |
| <else> | x | Parte de la estructura condicional <if> |
| <elseif> | Cond | Parte de la estructura condicional <if> |
| <enumerate> | x | Enumeración de las opciones de un menú |
| <error> | cond, count | Recoge el evento error |
| <exit> | expr, namelist | Sale de la sesión |
| <field> | cond, expr, modal, name, slot, type | Declara un campo de entrada en un formulario <form> |
| <filled> | mode, namelist | Acción a realizar cuando se rellena el campo <field> |
| <foreach> | array, ítem | Notación abreviada para un bucle <i>for</i> ² |
| <form> | id, scope | Formulario. Diálogo para presentar una información y recoger los datos. |

² Una estructura *for* es un bucle que repite una acción un número de veces determinado por el programador.

| | | |
|-------------|--|---|
| <goto> | expr, expritem, fetchaudio, fetchhint, fetchtimeout, next, maxage, maxstale, nextitem | Redirección a otro diálogo next dentro o fuera del documento. |
| <grammar> | fetchhint, mode, root, fetchtimeout, srcexpr, scope, src, tag-format, type, weight, version, voxeo:useuri, xml:base, xml:lang, xmlns | Indica la gramática o vocabulario para un diálogo. Puede ser de voz o DTFM |
| <help> | cond, count | Captura un evento de ayuda. |
| <if> | Cond | Estructura condicional si... entonces |
| <link> | dtmf, eventexpr, expr, event, fetchaudio, next, fetchhint, fetchtimeout, maxage, maxstale, message | Especifica un enlace común a todos los diálogos. |
| <log> | expr, label | Genera un mensaje de depuración |
| <menu> | dtmf, id, scope | Diálogo que oferta varias opciones |
| <noinput> | cond, count | Captura el evento de no entrada de datos |
| <nomatch> | cond, count | Captura el evento cuando no hay coincidencia entre la entrada y la gramática asociada |
| <one-of> | x | Permite construir la gramática con una serie de opciones |
| <option> | accept, dtmf, value | Especifica un opción en un <field> |
| <prompt> | bargein, bargeintype, cond, count, timeout, xml:lang | Reproduce un mensaje para el usuario |
| <record> | beep, cond, dtmfterm, expr, name, maxtime finalsilence, modal, type | Graba un mensaje vocal |
| <reprompt> | x | Reproduce el último <prompt> tras un evento |
| <return> | event, eventexpr, message, messageexpr, namelist | Vuelve desde un sub-diálogo |
| <script> | charset, fetchtimeout, src, srcexpr, maxage, fetchhint, maxstale | Enmarca un bloque de código ECMAScript para su ejecución. |
| <subdialog> | cond, enctype, src, expr, fetchaudio, fetchhint, name, namelist, maxage, maxstale, fetchtimeout, method | Llama a otro diálogo como subdiálogo del actual |

| | | |
|----------|--|---|
| <submit> | enctype, namelist, next, fetchaudio, fetchtimeout, method, fetchhint, expr | Envía valores de las variables a otros documentos alojados en un servidor |
| <throw> | event, eventexpr, message, messageexpr | Lanza un evento |
| <value> | Expr | Inserta el valor de de <i>expr</i> en un <prompt> |
| <var> | expr, name | Declara una variable de nombre <i>name</i> y contenido <i>expr</i> |
| <vxml> | application, version, xml:base, xml:lang, xmlns, xmlns:voxeo | Elemento raíz de cada documento VoiceXML |

Tabla 3 Elemento VXML –Atributo – Función

En la Tabla 4 se detallan las descripciones de algunos de los atributos que se han especificado en la tabla anterior. Estos atributos pueden tener funciones específicas dependiendo del elemento al que acompañen.

Dada la gran cantidad de atributos, sólo muestran aquellos más importantes, entre ellos los que se han utilizado en el proyecto que se describe en este documento.

| Atributo | Descripción |
|--------------|--|
| bargein | Indica si el usuario puede o no interrumpir un <prompt> |
| cond | Evalúa el valor de una variable o ítem |
| count | Indica cuántas veces se pueden producir distintos <prompt> si el usuario está repitiendo la misma acción |
| event | Evento: no-entrada, no-coincidencia... |
| expr | Valor inicial de una variable |
| fetchaudio | Especifica la URI de un archivo .WAV para su reproducción |
| fetchhint | Indica cuando una fuente (URI de audio, por ejemplo) debe estar lista |
| fetchtimeout | Indica por cuánto tiempo debe el intérprete intentar acceder al elemento |
| id | Nombre del elemento en cuestión. Se puede referenciar dentro del documento o fuera de él |
| name | Nombre de elemento o variable |
| namelist | Nombres de las variables o ítems. |

| | |
|----------|--|
| next | URI (nuevo documento o diálogo) al que dirigirse |
| scope | Ámbito, alcance del element. |
| src | URI que indica la fuente de un archivo |
| timeout | Tiempo de espera para lanzar un evento |
| value | Especifica el valor del elemento |
| version | Versión VoiceXML (obligatorio) |
| xml:base | URL base del documento. Todas las URIs relativas en el documento se derivan de esta. |
| xml:lang | Especifica el idioma de la aplicación. |
| xmlns | Espacio de nombres designado para VoiceXML (obligatorio) |

Tabla 4 Atributo – Descripción

Una vez conocidos los elementos y sus atributos, somos capaces de diseñar diálogos. Comenzaremos por definir los tipos de diálogos que existen: formularios y menús.

- **Formulario**

Los formularios son los elementos más importantes de un documento VXML. Marcan el guión por el cual se van recogiendo los datos. Además de la declaración de las variables, trata los eventos y ejecuta las acciones que sean pertinentes cuando se rellenan los campos de entrada.

Se guían por el algoritmo FIA (*Form Interpretation Algorithm*). La Figura 8 permite conocer en detalle el funcionamiento de este algoritmo:

1. Se declaran los campos o ítems de entrada `<field name=" "> [...] </field>` (línea 3).
2. Se incluyen las instrucciones precisas a realizar por el usuario y se le informa de ellas mediante `<prompt>` (línea 5).
3. Se validará la respuesta del usuario con la gramática asociada a él. (líneas 12 a 15).
 - a. Si no existe coincidencia o no hay entrada se lanzan los eventos `<noinput>` o `<nomatch>` respectivamente, y la

repetición del mensaje del usuario `<reprompt/>`. Así hasta que el usuario diga cualquiera de las siguientes palabras [película cartelera premios]

- b. Si el usuario ha dicho una palabra válida, se da por rellenado el campo `<field name=" ">` y por tanto pasa a ejecutar el bloque `<filled namelist=" ">` (línea 28) del mismo nombre que el campo. Todo el bloque `<filled>` es gran estructura condicional formada por las etiquetas `<if>`, `<elseif>` y su atributo `cond`. Es aquí donde se comprueba qué es lo que ha dicho el usuario, ya que, hasta ahora, solo sabíamos que había dicho *una* de las palabras clave. La respuesta del usuario se encuentra almacenada dentro de `Inicio`, y será la variable evaluada en `cond`. Según haya sido el resultado de la condición `Inicio` se pasará a la siguiente instrucción. Hace uso de otro elemento: `<goto>` y su propiedad `next` indica hacia dónde debe dirigirse cuando se cumpla la condición que precede a esta instrucción. Además se oirá un fragmento de audio mientras se accede al otro documento/diálogo (línea 30)

```

1  <!--[...]Formulario -->
2  <form id="MainMenu">
3    <field name="Inicio">
4      <audio src= "GG_stars.wav"/>
5    <prompt>
6      Bienvenido a MuviBox: el sistema de información de películas.
7      Para conocer información sobre una película en concreto diga:
8      película. Si desea conocer la cartelera, diga: cartelera. Para
9      saber las películas ganadoras de las últimas ediciones de
10     premios diga: premios.
11   </prompt>
12   <grammar version="1.0" xml:lang="es-es" type="text/gsl"
13     xmlns="http://www.w3.org/2001/06/grammar">
14     [película cartelera premios]
15   </grammar>
16
17   <noinput>
18     No he oído nada.
19   <reprompt/>
20 </noinput>
21
22   <nomatch>
23     No le he entendido.
24   <reprompt/>
25 </nomatch>
26 </field>
27
28 <filled namelist="Inicio">
29   <if cond="Inicio == 'película'">
30     <goto next = "#MovieForm" fetchaudio="TB_always_love.wav"/>
31   <elseif cond="Inicio == 'cartelera'"/>
32     <goto next = "#BillboardForm"/>
33   <elseif cond="Inicio == 'premios'"/>
34     <goto next = "#PrizesForm"/>
35   <else/>
36   </if>
37 </filled>
38 </form>

```

Fig. 8 Formulario VXML

- **Menú**

Con el menú (Figura 9) se le ofrece al usuario opciones entre las que debe elegir. Estas opciones están integradas dentro del elemento `<choice>`. Se trata de una versión más sencilla de un formulario.

```

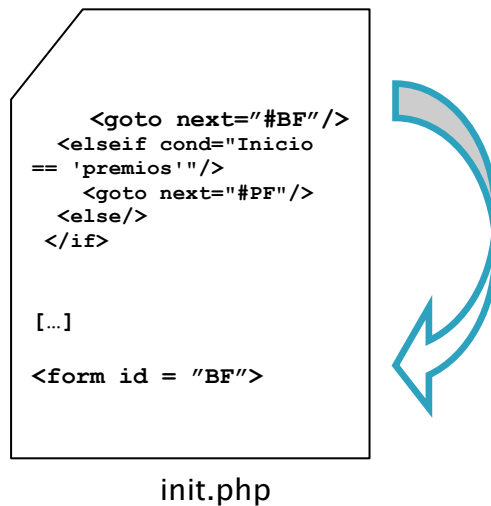
1  <!-- [...]Menu-->
2  <menu>
3      <prompt>
4          Bienvenido al inicio. Diga uno de: <enumerate/>
5      </prompt>
6      <choice next="http://www.deportes.ejemplo.com/vxml/inicio.vxml">
7          Deportes
8      </choice>
9      <choice next="http://www.tiempo.ejemplo.com/vxml/inicio_t.vxml">
10         Tiempo
11     </choice>
12     <choice next="http://www.noticias.ejemplo.com/vxml/intro.vxml">
13         Noticias astrofísicas
14     </choice>
15     <noinput> Por favor diga uno de: <enumerate/></noinput>
16 </menu>

```

Fig. 9 Menú VXML

Las opciones ofertadas en la Figura 9 son: Deportes, Tiempo o Noticias. En el formulario, serían el equivalente a las palabras que forman la gramática. En el ejemplo, la validación de la palabra dicha por el usuario y la acción a realizar se expresan dentro de la misma instrucción. Al elegir una de esas opciones automáticamente nos llevará al siguiente documento. Estas instrucciones se realizan en 2 pasos dentro de los formularios: primero se verifica que se ha dicho una de las palabras permitidas por la gramática, y después se evalúa cuál de ellas ha sido y se ejecuta la orden correspondiente. En un mismo documento pueden coexistir varios menús y varios formularios.

La Figura 10 muestra en detalle el funcionamiento de la instrucción `<goto>`. La instrucción de la línea 32 hace referencia a otro formulario dentro del mismo documento (línea 67). Cuando el sistema evalúe que la condición “Inicio” (línea 31) es “cartelera”, entonces el diálogo pasará al formulario identificado como “BillBoard”.



```

31 <elseif cond="Inicio == 'cartelera'"/>
32   <goto next = "#BillboardForm"/>
[...]
67 <form id="BillboardForm">
68 <field name="Billboard">
69   <prompt>
70     Elija el criterio para la cartelera. Si quiere saber
71     la cartelera, diga general. Para escuchar la lista
72     por orden de puntuación diga: puntuación. Para
73     conocer las más votadas diga: votos.
74   </prompt>
75   <grammar version="1.0" xml:lang="es-es" type="text/gsl"
76     xmlns="http://www.w3.org/2001/06/grammar">
77     .[general puntuación votos]
78   </grammar>
79
80   <noinput>
81     No he oído nada.
[...]

```

Fig. 10 Detalle <goto>

Los ejemplos anteriores han sido extraídos de su contexto original, donde existen las cabeceras del documento y el código completo.

Ahora que tenemos las estructuras necesarias para crear un diálogo, es necesario asociarle la gramática.

- **Gramática**

Conjunto de palabras que la aplicación va a considerar como aptas para ser pronunciadas por el usuario en un determinado momento de la interacción. El sistema reconoce qué gramática está

activa gracias al uso de flags³ que indican el diálogo y su gramática asociada. Existen dos tipos:

- Voz: el usuario se comunica de manera oral y son palabras lo que el sistema reconoce. Las palabras se especifican entre corchetes, y aquellas que han de pronunciarse juntas, a su vez, se agrupan entre paréntesis.

```
[ (los juegos del hambre) rec ]
```

Fig. 11 Declaración gramática – voz

- DTMF (dual –tone multi frequency): pulsaciones de tecla, el sistema reconoce el número que se ha pulsado. Se declara como `dtmf="#"`; donde # es un número.

Las gramáticas pueden declararse de manera local, es decir acompañando al elemento al que pertenecen (Figura 12) o bien, se pueden implementar en un fichero XML externo y ser llamado desde la aplicación. En `src` se indica la URI del fichero, en caso de encontrarse en el mismo directorio, basta con el nombre (Figura 13), donde `selInfo.xml` es el fichero mostrado en la Figura 14.

```
<grammar versión = "1.0" xml:lang = "es - es" type =  
"text/gsl" xmlns="http://www.w3.org/2001/06/grammar">  
[general puntuación votos] </grammar>
```

Fig. 12 Declaración gramática – local

```
<grammar src="selInfo.xml"></grammar>
```

Fig. 13 Declaración gramática – externa

```
1 <?xml version = "1.0"?>  
2 <grammar xml:lang="es-es" root = "selInfo"  
3 xmlns="http://www.w3.org/2001/06/grammar">  
4 <rule id="selInfo" scope = "public">  
5 <one-of>  
6 <item>sinopsis</item>
```

³ Un *flag* es una variable que se utiliza para señalar un estado concreto. Se trata de 1 o más bits que almacenan un valor binario.


```

7           <item>director</item>
8           <item>reparto</item>
9           <item>premios</item>
10        </one-of>
11    </rule>
12 </grammar>

```

Fig. 14 sellInfo.xml

Si la locución del usuario no concuerda con ninguna de las palabras incluidas en la gramática, o bien esta no ha dicho nada, se producen los eventos `<nomatch>` o `<noinput>`.

- **Evento**

Se trata de “acontecimientos especiales” que se encuentran fuera del comportamiento normal que se espera de la aplicación. Los eventos predefinidos en VoiceXML se resumen en la Tabla 5.

| | |
|--|--|
| <code><cancel></code> | El usuario ha pedido cancelar el prompt actual. |
| <code><connection.disconnect></code> | El usuario ha colgado. |
| <code><help></code> | El usuario pide ayuda. |
| <code><noinput></code> | El usuario no ha dicho nada. |
| <code><nomatch></code> | La aplicación no ha reconocido lo que el usuario ha dicho. |

Tabla 5 Eventos predefinidos en VoiceXML

Disponemos de todas las herramientas necesarias para crear un documento VXML y poder generar una aplicación.

- **Aplicación**

A nivel usuario, podemos llamar aplicación al producto final. Una definición más exacta sería el conjunto de documentos VoiceXML que tienen un mismo documento raíz. Una aplicación puede consistir únicamente en un único documento, en ese caso, él mismo será documento raíz (Figura 16). En el caso de existir varios, los

documentos “hijos” deben referenciar en su elemento `<vxml>` la aplicación raíz a la que pertenecen. (Figura 15). En la Figura 16 [BUI], el código de una aplicación con un documento raíz y un documento hijo.

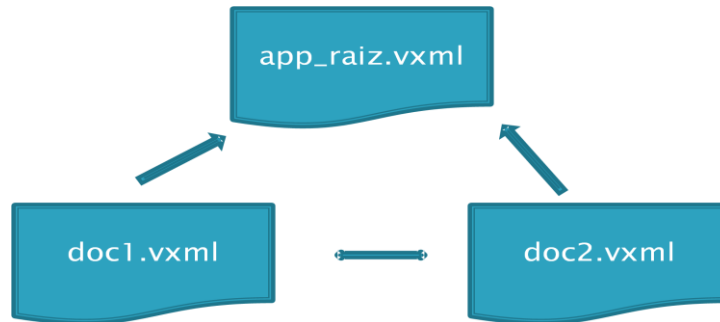


Fig. 15 Diagrama aplicación con documento raíz

```

1      <!-- app_raiz.vxml -->
2      <vxml version="2.1">
3          <var name="gVar" expr="1" />
4      </vxml>

1      <!--doc1.vxml -->
2      <vxml version="2.0" application="app_root.vxml">
3          <form id="form1">
4              <block>
5                  <assign name="gVar" expr="gVar+10" />
6              </block>
7          </form>
8      </vxml>

```

Fig. 16 Documento raíz y documento hijo

Para finalizar, ya podemos iniciar una sesión con nuestra aplicación.

- **Sesión**

Se inicia una sesión cuando el usuario accede a la aplicación y comienza en diálogo con el intérprete. Ésta sigue iniciada mientras los documentos estén cargados y finaliza por petición del usuario, o bien por iniciativa de la intérprete VoiceXML.

Los siguientes apartados se corresponden con otros estándares de desarrollo de sistemas de diálogo también recomendados por la W3C. Algunos de los citados a continuación actúan conjuntamente con VoiceXML.

2.3.2 Call Control eXtensible Markup Language (CCXML)

CCXML [CCXML] puede utilizarse como complemento de VXML para la gestión inicial de la llamada, para ello está soportado por algunas plataformas de VXML. La función de este lenguaje es mantener al corriente del estado de la llamada al servidor de voz, así como informar del manejo de la misma. Ambos estándares también pueden implementarse por separado.

2.3.3 Speech Recognition Grammar Specification y Semantic Interpretation for Speech Recognition (SRGS)

SRGS [SRGS] es un estándar que especifica el diseño de las gramáticas para el reconocimiento del habla del usuario. La sintaxis más extendida está basada en XML y trabaja conjuntamente con otro estándar de interpretación SISR (interpretación semántica para el reconocimiento del habla). [SISR] Mientras que el SRGS ha reconocido qué palabra ha dicho el usuario SISR le aporta el significado semántico necesario.

2.3.4 Speech Synthesis Markup Language (SSML)

[SSML] Se usa para manipular el habla del sintetizador, es capaz de modificar aspectos como el volumen, la entonación, pronunciación, etc. Mediante la iniciativa SABLE [SABLE] se trata de aunar todos los estándares destinados a la interpretación de los reconocedores del habla y sintetizadores de voz (SRGS, SISR...).



Capítulo 3

3 ESQUEMA GENERAL DE LA APLICACIÓN

A lo largo de esta sección se incluye una descripción general sobre la aplicación desarrollada en este Trabajo Fin de Grado y las herramientas que han sido necesarias en el proceso.

Como se ha venido diciendo a lo largo del documento, el presente trabajo versa sobre el desarrollo de una aplicación para informar a los usuarios sobre contenido cinematográfico.

3.1 Introducción al sistema

La aplicación se ejecuta dentro de un escenario cliente-servidor como el mostrado en la Figura 17.

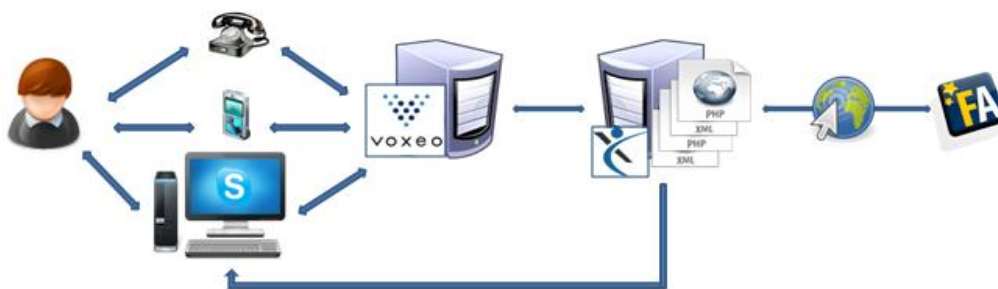


Fig. 17 Esquema cliente-servidor

El usuario desencadena la acción llamando al número de la aplicación. Esta llamada puede realizarse vía línea telefónica convencional, fija o móvil o bien desde un ordenador haciendo uso de la tecnología voz sobre Ip (Voip) accediendo desde, por ejemplo, Skype.

La solicitud de llamada llega al servidor de VoiceXML, en nuestro caso, se trata de la plataforma Voxeo [VOX]. Aquí únicamente está creada la aplicación de manera virtual, es decir, el fichero contenedor del código se encuentra en otro servidor. Voxeo nos sirve para enlazar con el primer diálogo de la aplicación que se comunica con el resto del sistema. Con la petición de información procesada, el diálogo se desplaza al resto de ficheros alojados en un servidor diferente. Durante la ejecución del código se recupera la información de FilmAffinity y se devuelve al usuario la información solicitada sin pasar de nuevo por la plataforma de Voxeo. La respuesta la recibe el usuario directamente del fichero desde donde se ejecutó la orden de búsqueda de información.

3.2 Herramientas

3.2.1 Intérprete de VoiceXML: Voxeo Evolution

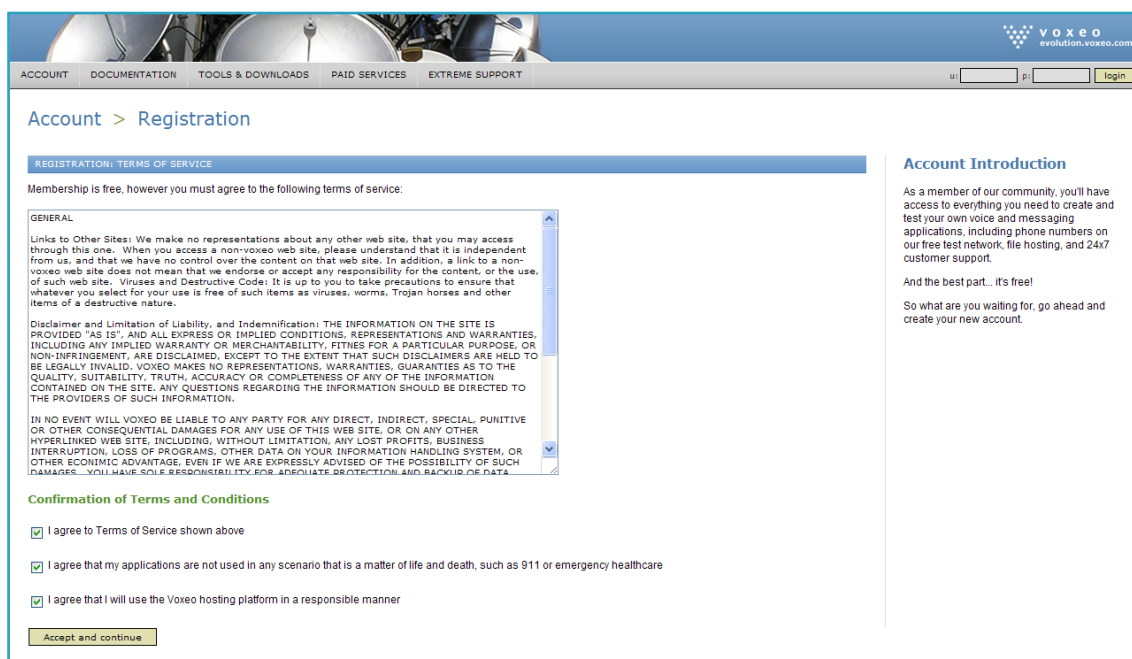
La compañía Voxeo proporciona una plataforma de desarrollo y alojamiento web para desarrollar aplicaciones de voz XML-IVR (*Interactive Voice Response*), recibe el nombre de *Evolution* [VOX]. Al registrar la aplicación se le asigna un número de teléfono para uso de la misma. Incluye una herramienta de depuración en tiempo real, permite el uso de VoIP y ofrece un soporte 24x7 a través de foros, además de numerosa documentación a la poder acceder en caso de dudas.

3 ESQUEMA GENERAL DE LA APLICACIÓN

El portal ofrece servicio gratuito de alojamiento para los archivos además de tener una interfaz muy sencilla y de fácil manejo. Otra característica de gran utilidad ha sido el sistema de depuración en tiempo real. Se activa al realizar una llamada y va siguiendo todos los pasos durante la realización de la misma. Esta herramienta ha sido muy utilizada para capturar fallos en momento de ejecución, ya que la mayoría de los mensajes de error mostrados son muy específicos sobre el error. A continuación se presenta una breve descripción de los pasos necesarios para acceder y manejar esta plataforma

Registro como usuario en Voxeo

Para poder comenzar a hacer uso de las herramientas, es necesario registrarse como usuario (Figura 18). Esta cuenta es totalmente gratuita, y aunque no da acceso al 100% de la documentación disponible, es más que suficiente para empezar. Acto seguido hay que elegir el estándar en la que vamos a desarrollar nuestra aplicación VoiceXML, CCXML, etc. En nuestro caso, se ha desarrollado la aplicación con VoiceXML.



The screenshot displays the Voxeo website's registration process. At the top, the navigation bar includes 'ACCOUNT', 'DOCUMENTATION', 'TOOLS & DOWNLOADS', 'PAID SERVICES', and 'EXTREME SUPPORT'. The main content area is titled 'Account > Registration' and features a 'REGISTRATION/ TERMS OF SERVICE' section. This section contains a 'GENERAL' heading and a scrollable area with the following text: 'Links to Other Sites: We make no representations about any other web site, that you may access through this one. When you access a non-voxeo web site, please understand that it is independent from us, and that we have no control over the content on that web site. In addition, a link to a non-voxeo web site does not mean that we endorse or accept any responsibility for the content, or the use, of such web site. Viruses and Destructive Code: It is up to you to take precautions to ensure that whatever you select for your use is free of such items as viruses, worms, Trojan horses and other items of a destructive nature. Disclaimer and Limitation of Liability, and Indemnification: THE INFORMATION ON THE SITE IS PROVIDED "AS IS", AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED. EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, VOXEO MAKES NO REPRESENTATIONS, WARRANTIES, GUARANTIES AS TO THE QUALITY, SUITABILITY, TRUTH, ACCURACY OR COMPLETENESS OF ANY OF THE INFORMATION CONTAINED ON THE SITE. ANY QUESTIONS REGARDING THE INFORMATION SHOULD BE DIRECTED TO THE PROVIDERS OF SUCH INFORMATION. IN NO EVENT WILL VOXEO BE LIABLE TO ANY PARTY FOR ANY DIRECT, INDIRECT, SPECIAL, PUNITIVE OR OTHER CONSEQUENTIAL DAMAGES FOR ANY USE OF THIS WEB SITE, OR ON ANY OTHER HYPERLINKED WEB SITE, INCLUDING, WITHOUT LIMITATION, ANY LOST PROFITS, BUSINESS INTERRUPTION, LOSS OF PROGRAMS, OTHER DATA ON YOUR INFORMATION HANDLING SYSTEM, OR OTHER ECONOMIC ADVANTAGE, EVEN IF WE ARE EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. YOU HAVE SOLE RESPONSIBILITY FOR ADEQUATE PROTECTION AND BACKUP OF DATA.' Below this, the 'Confirmation of Terms and Conditions' section has three checked checkboxes: 'I agree to Terms of Service shown above', 'I agree that my applications are not used in any scenario that is a matter of life and death, such as 911 or emergency healthcare', and 'I agree that I will use the Voxeo hosting platform in a responsible manner'. An 'Accept and continue' button is at the bottom. On the right, the 'Account Introduction' section welcomes the user and mentions access to a free test network, file hosting, and 24x7 customer support. A 'login' button is located in the top right corner.

Fig. 18 Creación de cuenta en Voxeo

3 ESQUEMA GENERAL DE LA APLICACIÓN

Una vez que ya somos usuarios registrados y podemos acceder con nuestra cuenta, tenemos a nuestra disposición numerosas herramientas para facilitar el desarrollo de la aplicación (Figura 19).

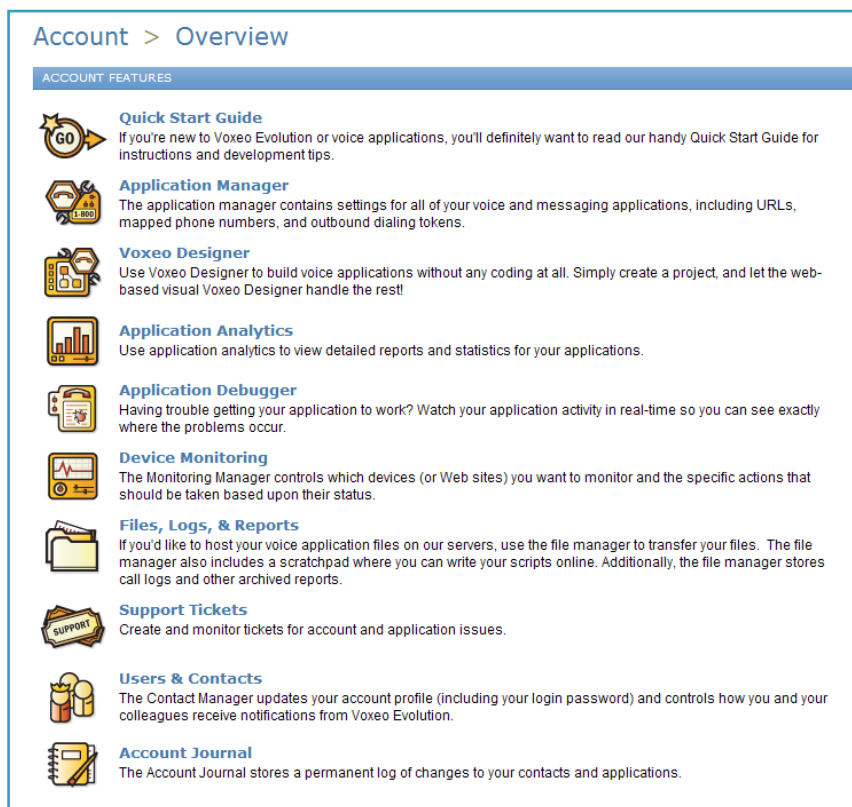


Fig. 19 Detalle vista general cuenta Voxeo

Se van a describir a continuación aquellas de las que se ha hecho uso durante el desarrollo del presente proyecto:

- **Quick Start Guide:** guía rápida para nuevos usuarios.
- **Application Manager:** gestor de aplicaciones, contiene la configuración para todas las aplicaciones desarrolladas por el usuario registrado, incluye las URL, el alojamiento de los archivos, etc.
- **Application Debugger:** herramienta de gran utilidad para depurar los documentos de la aplicación en tiempo real. Se suceden los mensajes que indican por dónde va la aplicación, qué valores toman las variables... Muy eficiente para analizar un error interno.

3 ESQUEMA GENERAL DE LA APLICACIÓN

- **File, Logs and Reports:** es el alojamiento web de los documentos y archivos VXML o de gramáticas (.jsgf).
- **Support Tickets:** se pueden abrir *tickets* (hilos) para consultar dudas en los foros de soporte, las respuestas ofrecidas por el personal resulta de gran ayuda.

Registro de la aplicación

La pantalla de registro de la aplicación se muestra en la Figura 20.

ACCOUNT DOCUMENTATION TOOLS & DOWNLOADS PAID SERVICES EXTREME SUPPORT

Account > Applications > AApp_Ini

APPLICATION SETTINGS

Application Settings Contact Methods

* Application Name: AApp_Ini

* What forms of communication will this application support?

☒ Voice phone calls
☐ Text messaging
☐ Both

* Voice Application Type:

| Deployment | Region | App Type | ASR/TTS | Platform |
|-------------|--------|--|---|----------------------|
| Development | USA | CCXML CallXML Designer VoiceObjects VoiceXML | DTMF-Only Nuance 9 Premium ASR Premium ASR/TTS | Prophecy 11 VoiceXML |

Selected application type: Staging, Prophecy 11 VoiceXML, Premium ASR/TTS

* Voice URL: file manager | view file
http://tfg.x10.mx/init.php

+ Add a failover URL

Update Application Delete Application

Fig. 20 Registro aplicación Voxeo

Observaciones:

- Región ¿Por qué USA?: actualmente, no se están ofertando números de Skype para Europa, por tanto se registró la aplicación en Estados Unidos. Esta selección no influye para nada en la interacción con la misma.
- Voice URL: como se ve en la figura, el dominio no corresponde al de la plataforma Voxeo, efectivamente, el documento principal se encuentra en un servidor externos por dos razones:
 1. Se trata de un archivo PHP, extensión no soportada por Voxeo Evolution [VOX].

3 ESQUEMA GENERAL DE LA APLICACIÓN

2. Para facilitar la sintaxis y la invocación de otros documentos, se decidió alojar en el mismo servidor.

En la pestaña *Contact Methods* se encuentran los números de teléfono asignados a la aplicación. Tal y como se observa en Figura 21, el número de telefonía fija asignado pertenece a la Comunidad de Madrid.

Account > Applications > AApp_Ini

APPLICATION SETTINGS

Application Settings | **Contact Methods**

Phone Numbers & Addresses

The following contact numbers and addresses are mapped to your application.

| <input type="checkbox"/> Number Type | Number |
|---|-------------------------------------|
| <input type="checkbox"/> International (Voice Only) - Spain | +34 91 1235303 |
| <input type="checkbox"/> INum Number (Voice Only) | +883510001818727 |
| USA Toll Free PIN Access (Voice Only) | (800) 289-5570 then PIN: 9990083286 |
| USA Domestic PIN Access (Voice Only) | (407) 386-2174 then PIN: 9990083286 |
| Skype VoIP | +990009369990083286 |
| SIP VoIP | sip:9990083286@sip.voxeo.net |
| Phono Number | app.9990083286 |

Use the following form to add a new number to this application. Please note that international numbers may have [country-specific restrictions](#).

Phone Number:

United States (+1)

Fig. 21 Información de contacto

3.2.2 Servidor de alojamiento web: x10hosting

La necesidad de tener otro servidor externo, aparte de la plataforma Voxeo, surge a raíz de la incompatibilidad de Voxeo con el lenguaje PHP. Éste, al igual que ASP o JSP no son soportados por la plataforma Evolution [VOX]. No obstante, este hecho no presenta ningún problema a la hora de crear una aplicación cuyo fichero inicial o principal se encuentre fuera de Voxeo, basta con añadirla cuando va a crearse, tal y como muestra la Figura 22.

* **Voice URL:** [file manager](#) | [view file](#)

[+ Add a failover URL](#)

Fig. 22 Enlace a fichero inicial desde Voxeo

3 ESQUEMA GENERAL DE LA APLICACIÓN

El desconocimiento inicial sobre PHP en general y los servicios de hosting para ficheros, provocaron algunos problemas con la aplicación. Los ficheros PHP eran incapaces de alcanzar la página web de *FilmAffinity* desde el servidor, sin embargo, ejecutados en modo local devolvían la información correctamente. Con un estudio más detallado del problema, se obtuvo el origen del problema:

Existe un parámetro que hay que habilitar para permitir el acceso a objetos de tipo URL, páginas web; por defecto, viene deshabilitado en la mayoría de servidores por motivos de seguridad. Se trata de la opción *allow_url_fopen* y la sintaxis para la habilitación es la siguiente: *allow_url_fopen=on*, aunque no en todas las configuraciones de los servidores es editable, como ya se ha dicho, en la mayoría viene deshabilitada y sin opción a cambio.

Por tanto, la principal motivación para la elección de este servidor fue que permitía el acceso a webs desde él mismo. Además existen otras razones por las que la selección del servidor web x10hosting ha resultado de gran utilidad en el transcurso del desarrollo de la aplicación.

- Uso intuitivo y multitud de operaciones posibles:

- **cPanel**

El panel de control tiene un diseño sencillo que hace posible no perder el tiempo en el manejo de la interfaz (Figura 23).

3 ESQUEMA GENERAL DE LA APLICACIÓN

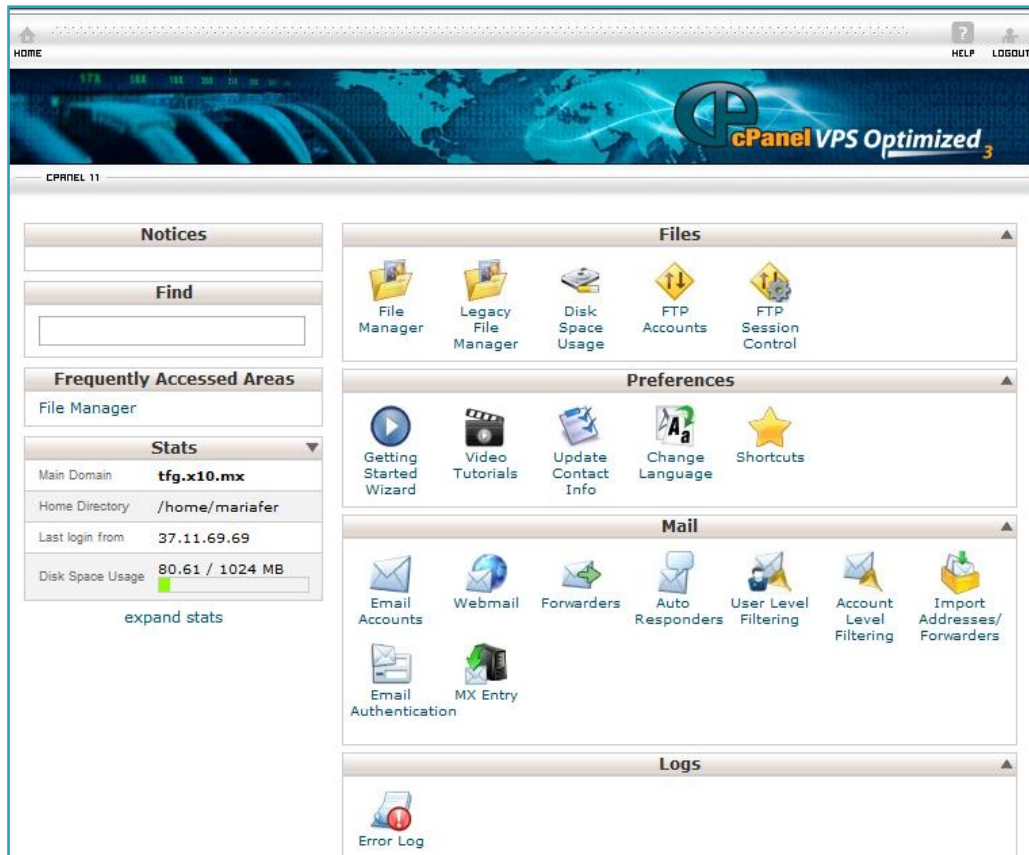


Fig. 23 Panel de control x10hosting

- **File Manager**

El administrador de ficheros, mostrado en la Figura 24, permite gran cantidad de operaciones. Aparte de las más usuales tales como subir o descargar ficheros, podemos encontrar la opción de comprimir los ficheros en una carpeta, renombrarlos, *etc.* Otra de las propiedades de este servidor es que ofrece la posibilidad de edición de ficheros *online* y con opción de resaltar con colores las etiquetas de código. Esto resulta de gran utilidad en la programación, ya que facilita saber qué puede faltar y distinguir visualmente las palabras reservadas.

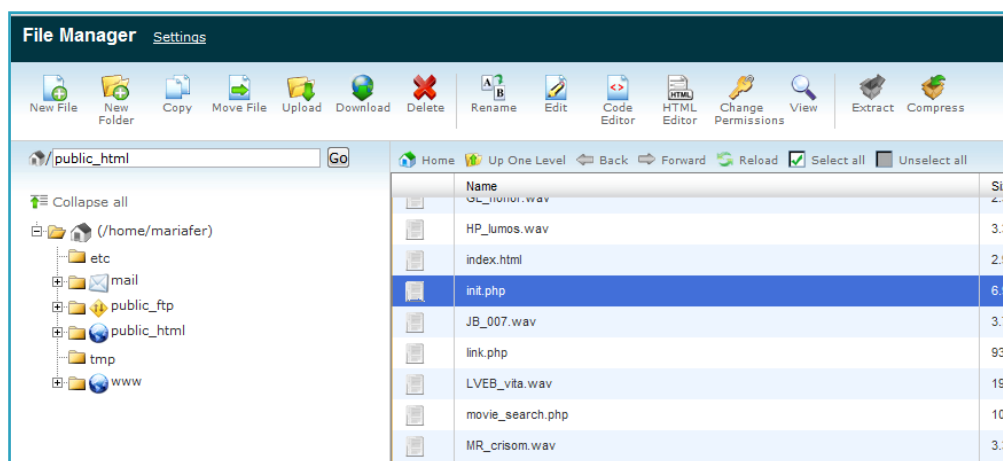


Fig. 24 Administrador de ficheros x10hosting

- **Hosting gratuito**

Se trata de un servidor de alojamiento gratuito que además nos ofrece soporte para PHP agregando el permiso para habilitar el acceso a páginas web, hasta 2 bases de datos SQL, que si bien ahora no se están utilizando, no se descartarían en un futuro.

3.2.3 Obtención de la información: FilmAffinity

Como se ha mencionado anteriormente, este proyecto carece de base de datos. La información se recoge directamente de la página web FilmAffinity. Una de las mayores bases de datos sobre cine en español en estos momentos.

Aunque la web no se pensó inicialmente como una página de consulta únicamente, sino que la idea de crear una comunidad donde la gente pudiera participar y dar su opinión y valoración fue una de las principales motivaciones. Se trata de un sistema donde los usuarios registrados evalúan las películas y aportan su opinión. Basándose en estas puntuaciones, aquellos usuarios que han mostrado más afinidad en sus valoraciones quedan “marcados” como *almas gemelas*. Se trata de un servicio para usuarios registrados, y de momento, la aplicación sólo accede a la información pública. No obstante, para futuras versiones podría ser desarrollado un diálogo previo para la identificación del usuario.

3.3 Módulos principales

La Figura 25 muestra los módulos fundamentales de la aplicación desarrollada.

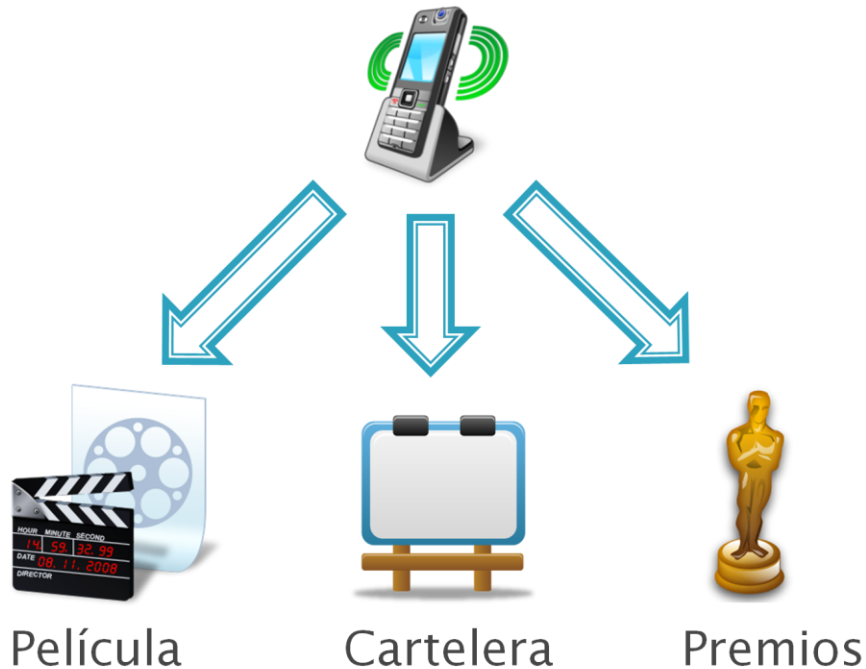


Fig. 25 Esquema general de la aplicación

La información que aporta el sistema se divide en varios módulos principales:

🎬 Película

Con el título de la película podemos conocer:

- Sinopsis.
- Director.
- Reparto.
- Premios.

🎬 Festivales

Dado el nombre del festival o el de la ceremonia de entrega de premios podemos conocer las películas ganadoras en el último año:

- Cannes.
- Venecia.
- Berlín.
- San Sebastián.
- Toronto.
- Oscar.
- Goya.
- Globos de Oro.

Cartelera

El sistema informará de las 10 primeras películas que hay en ese momento en cartelera:

- Vista general
- Por número de votos
- Por votación

La Figura 26 resume las funcionalidades descritas, que se detallan a continuación en las siguientes subsecciones de la memoria.

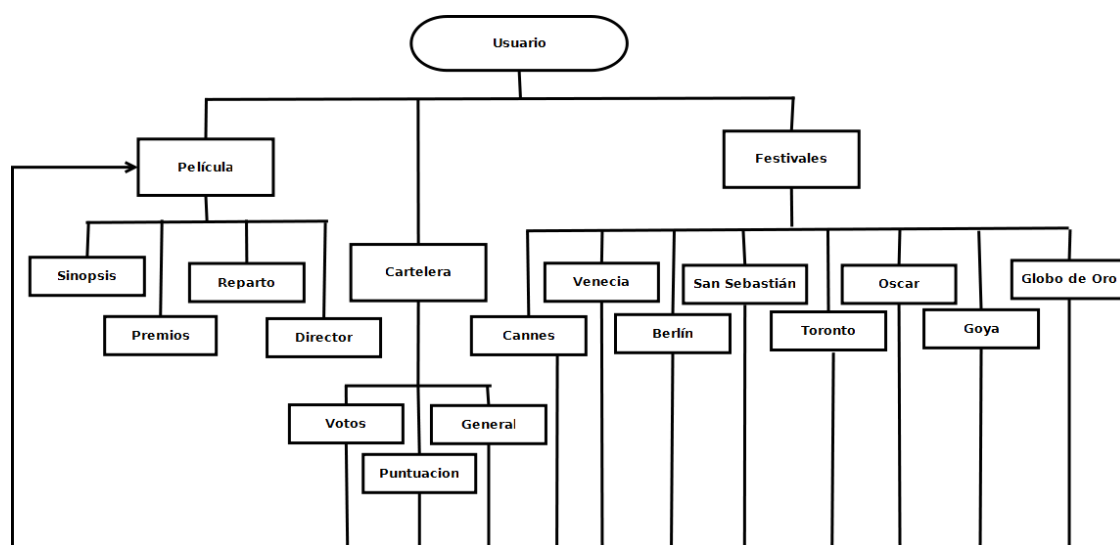


Fig. 26 Diagrama organización aplicación

3.3.1 Película

Se le pide al usuario que proporcione el título de la película, inicialmente se dispone de un número reducido de películas, pudiendo aumentar en cualquier momento. En caso de que exista coincidencia con más de uno se le volverá a pedir que concrete sobre el título de su búsqueda. Una vez se tiene un único resultado en la búsqueda se vuelve a pedir la intervención del usuario para saber qué tipo de información desea obtener.

Como se ha detallado en el apartado anterior, las posibles informaciones son: la sinopsis de la película, el reparto, el director o si tiene premios. Acto seguido, preguntará al usuario si quiere conocer

más sobre la película, si desea volver al menú inicial o bien salir de la aplicación.

3.3.2 Festivales

Si el usuario ha elegido la opción festivales, será necesario dar el nombre del festival o premio del que queremos conocer el ganador. Una vez conocida la película ganadora se ofrece al usuario la posibilidad de conocer más sobre esta película. En concreto, se puede saber de qué trata, quién la ha dirigido, quiénes son los actores y qué premios ha recibido. Al igual que en el módulo anterior, al finalizar se informará al usuario de las posibilidades de las que dispone a continuación.

3.3.3 Cartelera

Es el módulo más dinámico de los tres. Se actualiza cada semana con los estrenos y por tanto, la información que se aporta al usuario es nueva cada vez que haya un nuevo estreno. Esto se debe a la recuperación de información de manera directa de la web de FilmAffinity. Gracias a esto, podemos disponer de información de más películas sin necesidad de incluirlas en la lista principal. Tal y como se ha descrito anteriormente, al concluir el dictado de la información se procede de la misma manera que en los casos anteriores.



Capítulo 4

4 DISEÑO DE LA APLICACIÓN: MovieBox

4.1 Requisitos

Para poder hacer uso de la aplicación únicamente es necesario un terminal telefónico, ya sea móvil o fijo, con su correspondiente línea. O bien, también puede accederse a ella a través de VoIP (voz sobre IP) haciendo uso de Skype, gracias al acuerdo alcanzado en 2008 por ambas compañías: Voxeo y Skype [VOX-SKY].

4.2 Fases de desarrollo

4.2.1 Fase previa

- Estudio de las tecnologías y herramientas de desarrollo

Al comenzar este proyecto desconocía, más allá de nivel usuario, los sistemas de diálogo. Para implicarme como

programadora en el desarrollo de una aplicación de estas características necesitaba profundizar en estos conocimientos.

Dediqué las primeras semanas al estudio de los sistemas de diálogo, sus posibilidades, y sobre todo el lenguaje de desarrollo: VXML. Gracias a las páginas proporcionadas por el tutor del trabajo, encontré tutoriales muy completos que cubrían ampliamente cualquier duda que pudiese tener, a priori, sobre la sintaxis del código así como numerosos ejemplos.

A la vez que iba estudiando en la programación de este lenguaje, me iba familiarizando con la plataforma de desarrollo y alojamiento de los ficheros de la aplicación.

Para creación de aplicaciones complejas basadas en sistemas de diálogo no basta únicamente con VXML, es necesaria la implementación de funciones más complicadas que precisan de lenguajes de más alto nivel. En este caso el lenguaje elegido fue PHP. Aunque no había estudiado previamente este lenguaje, dada su similitud con otros aprendidos durante la carrera, no fue una tarea difícil adaptarse a él.

- **Delimitación de las funcionalidades de la aplicación**

Uno de los pasos clave previos al desarrollo de la aplicación es fijar bien qué servicios se quieren ofrecer con el sistema. Decidir de antemano las funciones del proyecto ayuda a concentrar esfuerzos en el desarrollo y perfeccionamiento de los mismos: qué tipo de información iba a ofrecérsele al usuario, qué poder de decisión tendría el usuario, etc. Para ello se estudió el acceso de los usuarios a webs sobre la misma temática que este trabajo, con el fin de seleccionar qué información se requería con mayor frecuencia.

Tan importante como las funciones de la aplicación es la fuente de dónde obtener la información. Tomada la decisión de no precisar de base de datos, había que elegir una fuente fiable, de fácil acceso y manejo.

4.2.2 Fase de desarrollo

Para empezar con el desarrollo del sistema y siguiendo los consejos de mi tutor, comencé por dar forma a la aplicación mediante modelos simplificados de lo que sería el prototipo final.

- **Paso 1: diseñar el sistema de forma estática**

Superado el nivel de manejo de variables dentro del mismo script, el siguiente escalón consistía en separar los códigos en dos ficheros distintos (Figura 27). De esta forma podía comprobar que era capaz de recuperar información que se encontraba fuera de mi archivo principal. De momento solo se pedía información sobre una película, siendo unívoca la respuesta. El diálogo secundario, bien podría ejecutarse desde el archivo raíz o bien desde el que contiene la fuente de información

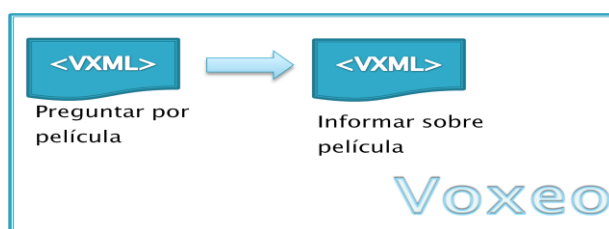


Fig. 27 Prototipo estático de la aplicación

- **Paso 2: alojar el fichero con la información solicitada a un servidor web externo**

De esta forma se verifica la correcta comunicación entre la plataforma Voxeo y el servidor web escogido (x10hosting). Ahora se incluyen más películas, por lo que entra en juego el reconocedor del habla de la plataforma Voxeo (Figura 28).

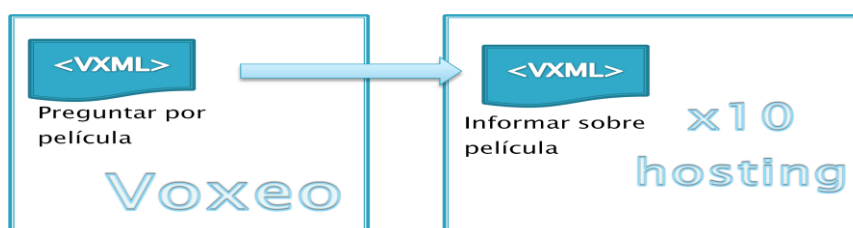


Fig. 28 Prototipo con servidor externo

- **Paso 3: recuperar información de forma dinámica.**

Se descarta la idea de manejar bases de datos y la de tener multitud de archivos con la información requerida. La idea es obtener la información en tiempo real dinámicamente. Para ello se hace uso del lenguaje PHP y la tarea consiste en desarrollar el código necesario para poder recuperar la información de una página externa y ajena a mi sistema, así como compatibilizarlo con el lenguaje VXML (Figura 29).

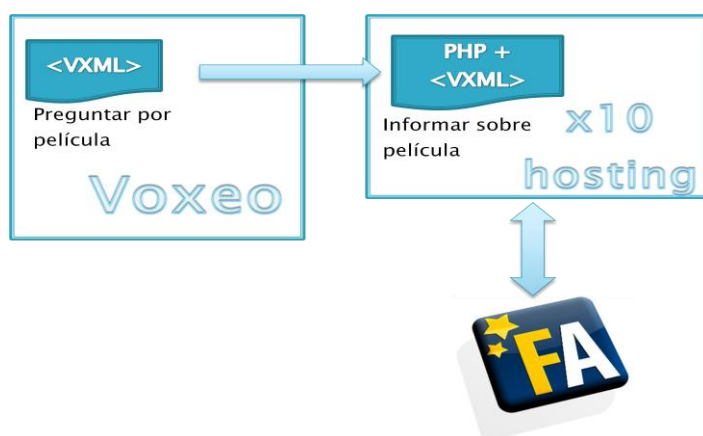


Fig. 29 Prototipo dinámico de la aplicación

4.3 Módulos

En total, la aplicación está compuesta por 6 ficheros, se pueden dividir en tres grupos:

- **Módulo inicial: init.php.**

Es el acceso a la aplicación, y aunque tenga extensión PHP, únicamente contiene instrucciones en VXML.

- **Módulo principal**

Incluye los tres módulos principales sobre los que versa la aplicación.

- **Módulo auxiliar**

Contiene ficheros auxiliares mayoritariamente programados en PHP, y no establece ningún diálogo con el usuario. Está formado por dos archivos.

- **functions.php**

Contiene las funciones comunes al resto de módulos. Se podría asemejar a las llamadas librerías en los entornos de programación.

- **link.php**

Se trata de un fichero de enlace para devolver información a otro fichero (movie_search.php).

4.3.1 Módulo inicial

Es el punto de acceso a la aplicación, init.php. Contiene el diálogo que nos da la bienvenida y nos informa de las primeras acciones que podemos realizar. Aunque únicamente está programada en lenguaje VXML, para mantener todos los archivos en el mismo servidor y con la misma estructura se almacenó como PHP, ya que son completamente compatibles.

Cuenta con pequeños fragmentos de música escogidos de las bandas sonoras más representativas del cine, para amenizar la espera entre los intercambios de ficheros. Al inicio escucharemos parte del tema que compuso John Williams para La Guerra de las Galaxias, también oiremos el tema principal para El Padrino, parte de El Guardaespaldas, etc.

Este fichero se divide en formularios como los descritos en secciones anteriores:

→ `<form id="MainMenu">`

TTS: Bienvenido a MovieBox: el sistema de información de películas. Para conocer información sobre una película en concreto diga: película. Si desea conocer la cartelera, diga: cartelera. Para saber las películas ganadoras de las últimas ediciones de premios diga: premios.

Fig. 30 Bienvenida al sistema MovieBox

Como ya se ha explicado, este tipo de aplicaciones requieren una gramática para poder interactuar con los usuarios. En este caso la gramática está compuesta por las palabras: película, cartelera o premios.

Cada una de ellas da paso a un nuevo formulario donde continuar recogiendo los datos necesarios para la consulta.

→ `<form id="MovieForm">`

Si se ha elegido la opción “película” oiremos la frase mostrada en la Figura 3.

TSS: *Diga el nombre de la película.*
Usuario: *Carros de Fuego*

Fig. 31 Diálogo petición título

A continuación se espera recibir el título de la película por parte del usuario. Inicialmente se dispone de un número limitado de posibles títulos incluidos en la gramática. Como curiosidad, añadir que aquellas películas cuyo título no se tradujo, han de pronunciarse en una versión “españolizada” y por tanto la gramática debe adaptarse. La lista de posibles nombres de películas es la siguiente:

Los juegos del hambre, Rec, Sombras Tenebrosas Descubriendo nunca jamás, Ira de titanes, La novia cadáver, El Rey León, Gladiator (pronunciación: *gladieitor*), Carros de fuego, Harry Potter (*jarry poter*), La Guerra de las Galaxias, Marea Roja, The Guardian (*de guardian*), El guardaespaldas.

Una vez elegida la película, a través de la etiqueta `<submit>` se pasa el título de la película a otro fichero PHP: *movie_search.php*; que se encarga de su procesado (Figura 32).

```
<submit next="movie_search.php" method ="get" namelist="titulo"[...]/>
```

Fig. 32 Detalle <submit> movie_search

→ <form id="BillboardForm">

Si hemos decidido conocer la cartelera, ésta se nos presenta con diversas opciones (Figura 33).

TSS: *Elija el criterio para la cartelera. Si quiere saber la cartelera, diga general. Para escuchar la lista por orden de puntuación diga: puntuación. Para conocer las más votadas diga: votos.*

Fig. 33 Diálogo petición cartelera

A continuación la gramática asociada y su significado:

- General: simplemente nos dice la lista de películas que se encuentran esa semana en cartelera, sin ningún orden especial. Cada una de las películas se acompaña de un número para poder elegirla y saber más cosas de ella.
- Puntuación: la comunidad de FilmAffinity se caracteriza por la gran participación de sus usuarios. Desde la administración de la página se anima a los usuarios a votar las películas para orientar a otros visitantes. Esta opción de cartelera nos viene ordenada según estos puntos que se dicen acompañando al título de la película, junto al número identificativo (Figura 34). Por ejemplo:

TSS: *Para Intocable valorada con un 8,1 diga 1*

Fig. 34 Diálogo película y valoración

–Votos: la primera película será aquella que haya sido votada más veces. No necesariamente han de ser votos positivos, podría ser equivalente a la película más vista.

Según el criterio escogido, se llamará al código billboard_search.php, tal y como muestra la Figura 35 para que gestione la respuesta.

```
<submit next="billboard_search.php" method ="get" namelist="vista" [...]"/>
```

Fig. 35 Detalle <submit> billboard_search.php

→ <form id="PrizesForm">

Oiremos el menú de la Figura 36 si la opción elegida es premios. Al decir cualquiera de los nombres de los festivales o premios anteriores, oiremos el nombre de la película ganadora, así como un número para poder elegirla (Figura 36) y saber más, tal y como se describió para el caso de la cartelera (Figura 37).

TSS: *Diga el nombre del festival si desea conocer la película ganadora de: Cannes, Venecia, Moscú, Berlín, Toronto o San Sebastián. Elija uno de los premios para saber qué película ganó el: Oscar, Globo de Oro o Goya. Si quiere saber qué película es la preferida por los usuarios de FilmAffinity, diga: usuarios.*

Fig. 36 Diálogo petición nombre del festival

```
<submit next="prize_search.php" method ="get" namelist="premio cat" [...]/>
```

Fig. 37 Detalle <submit> prize_search.php

4.3.2 Módulo principal

movie_search.php

Combina los dos lenguajes, PHP y VoiceXML. El primero de ellos se utiliza recuperar el título de la película y almacenar en variables las posibles consultas del usuario. Para poder responder al usuario es necesario el lenguaje VXML y la utilización de etiquetas especiales para poder trasladar valores de un lenguaje a otro, tal y como muestra la Figura 38.

```

1  <?php
2      $title = "''.normalize($results[0][9]).'";
3      $id = $results[1][9];
4  ?>
5  <var name="id" expr="<?=$id?>" />
6  <var name="tit" expr="<?=$title?>" />

```

Fig. 38 Utilización de PHP conjuntamente con VoiceXML

Entre las líneas 1 y 4 se le asigna un valor a las variables \$title e \$id en lenguaje PHP. En las siguientes, se recupera ese valor para ser usado en VXML.

Una vez recuperado el título de la película se ejecutan todas las funciones necesarias para disponer de la información que el usuario pueda precisar. Estas operaciones se hacen a priori para hacer más fluida la interacción con el usuario en ese diálogo. Recordemos que al llegar hasta aquí, la aplicación únicamente dispone del título de la película.

El flujo de información circula tal y como muestra la Figura 39.

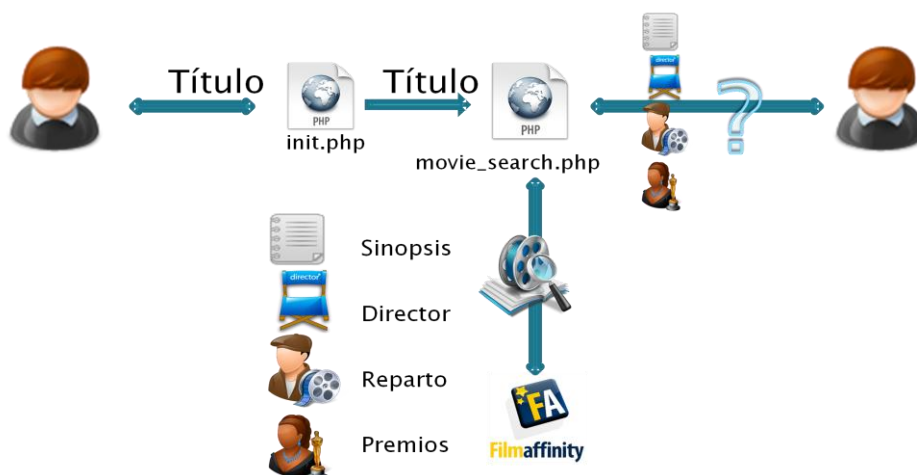


Fig. 39 Flujo información general movie_search

En la primera interacción con el usuario, en el inicio de la aplicación (init.php) se le ofrece la posibilidad de elegir decir el título de una película para saber información sobre ella. Acto seguido se le

pide el título de la misma y a partir de ahí la información pasa al módulo *movie_search.php*.

1. Para obtener el nombre de la película se hace uso de la instrucción REQUEST.

```
$movieTitle = $_REQUEST4["titulo"];
```

2. Conseguido el título se procede con la búsqueda. En esta ocasión pueden darse dos situaciones:

- a. Que la búsqueda obtenga únicamente un resultado. En este caso, no hay problema, se recupera la información sobre la sinopsis, el director, el reparto y los premios de la película.
- b. Si la búsqueda obtiene múltiples resultados, se le preguntará al usuario en qué película de todas las posibles coincidentes está interesado. Conocido el título concreto se inicia de nuevo el proceso, esta vez con un parámetro nuevo.

Pongamos un ejemplo para clarificar estos supuestos (Figuras 40 y 41).

a.

| | |
|----------|--|
| TTS | <i>Diga el nombre de la película</i> |
| Usuario: | <i>Sombras Tenebrosas</i> |
| TTS:: | <i>¿Qué quiere saber de Sombras Tenebrosas?</i> <i>Diga: sinopsis, director, reparto o premios.</i> |

Fig. 40 Resultado de búsqueda de título: un único resultado

b.

| | |
|----------|--|
| TTS: | <i>Diga el nombre de la película</i> |
| Usuario: | <i>Los Juegos del Hambre</i> |
| TTS: | <i>Hay más de un resultado, tiene que concretar la búsqueda: Para Los Juegos del Hambre: Sinsajo, parte 2, diga 1. Para Los Juegos del</i> |

⁴ REQUEST recoge el valor de la variable que le envía la instrucción complementaria GET. Son métodos para recuperar información en el lenguaje PHP.

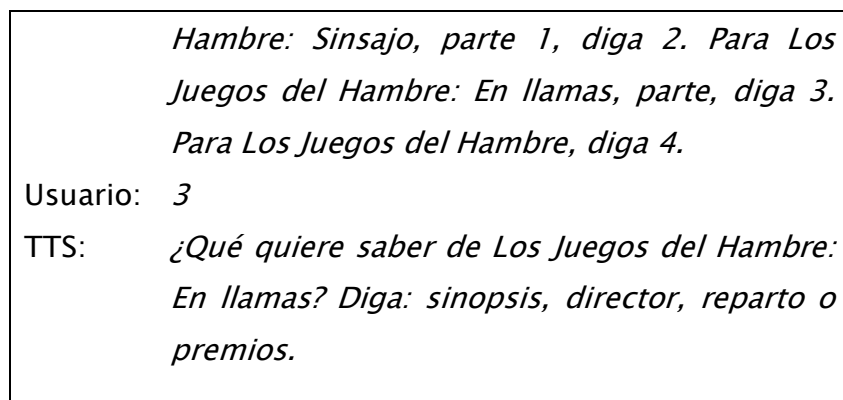


Fig. 41 Resultado de búsqueda de título: más de un resultado

En el supuesto b. la elección de la película queda claramente identificada por el título, por tanto, no habría confusiones a la hora de recuperar la información. Sin embargo, ¿cómo se recuperaría la información en el caso de haber elegido la opción 4: Los juegos del Hambre?

Para aclarar conceptos, haber elegido la opción “Los Juegos del Hambre” y volver a iniciar todo el proceso de búsqueda hubiera dado los mismos resultados que la primera vez, múltiples coincidencias.

Para evitar estas situaciones se le indica al usuario que diga el número que acompaña al título de la película. Mediante esta técnica nos aseguramos que no se trata de una nueva búsqueda, sino de una película en concreto. Internamente se almacenará el número de identificación único de la película, dado por FilmAffinity⁵; para conseguir la información relevante. El diagrama de la Figura 42 muestra el proceso descrito anteriormente.

⁵ <http://www.filmaffinity.com/es/film343660.html>

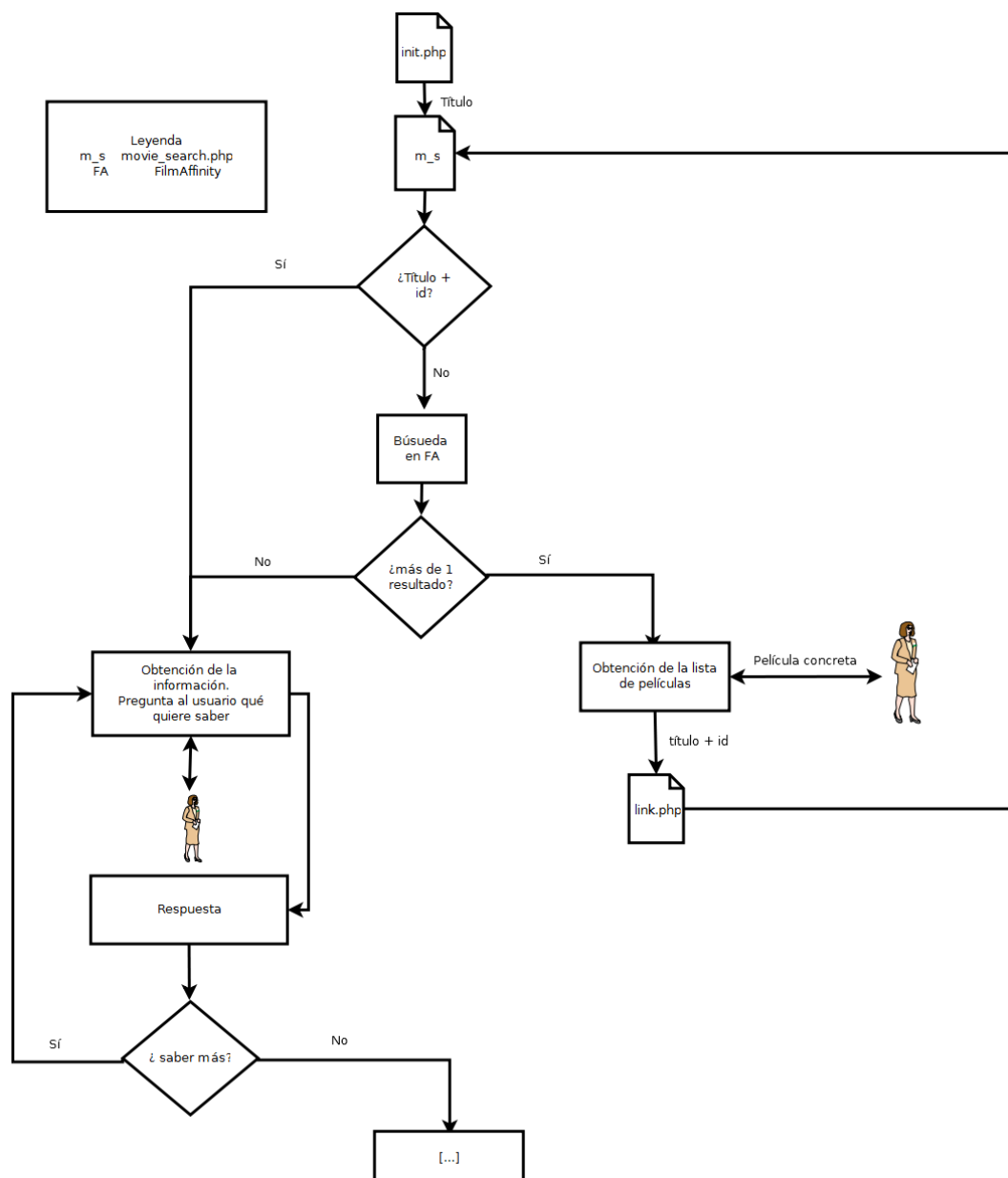


Fig. 42 Diagrama de bloques movie_search

billboard_search.php

En este dialogo obtenemos la lista de las 10 últimas películas estrenadas en España. Como ya se mencionó anteriormente, podemos obtener las películas ordenadas de distintas formas. Después de informar sobre la lista se ofrece la posibilidad de conocer más sobre la película, devolviendo la conversación al archivo movie_search.php.

4 DISEÑO DE LA APLICACIÓN: MovieBox

A continuación, se muestran unos ejemplos de cómo sería el diálogo Figuras 43 y 44.

| | |
|----------|--|
| TSS: | <i>[...]Si desea conocer la cartelera, diga: cartelera. [...]</i> |
| Usuario: | <i>Cartelera</i> |
| TSS: | <i>Elija el criterio para la cartelera. Si quiere saber la cartelera, diga general. Para escuchar la lista por orden de puntuación diga: puntuación. Para conocer las más votadas diga: votos.</i> |
| Usuario: | <i>General</i> |
| TSS: | <i>Para El legado de Bourne, diga 1. Para Piratas, diga 2. Para Los tres chiflados diga 3. Para Café de Flore, diga 4. Para Quiero ser italiano, diga 5 [...]</i> |
| Usuario: | <i>5</i> |
| TSS: | <i>¿Qué quiere saber de Quiero ser italiano? Diga sinopsis, director, reparto o premios.</i> |
| Usuario: | <i>Director</i> |
| TSS: | <i>Olivier Baroux. ¿Quiere saber algo más de Quiero ser italiano, diga si o no [...]</i> |

Fig. 43 Diálogo Cartelera General (15 Agosto 2012)

| | |
|----------|--|
| TSS: | <i>[...]Si desea conocer la cartelera, diga: cartelera. [...]</i> |
| Usuario: | <i>Cartelera</i> |
| TSS: | <i>Elija el criterio para la cartelera. Si quiere saber la cartelera, diga general. Para escuchar la lista por orden de puntuación diga: puntuación. Para conocer las más votadas diga: votos.</i> |
| Usuario: | <i>Puntuación</i> |
| TSS: | <i>Para La Luna valorada con 7,8 diga 1. Para El Caballero Oscuro La Leyenda Renace valorada con un 7,7 diga 2 [...]</i> |
| Usuario: | <i>2</i> |
| TSS: | <i>¿Qué quiere saber de El Caballero Oscuro La Leyenda Renace? Diga sinopsis, director, reparto o premios.</i> |
| Usuario: | <i>Premios</i> |
| TSS: | <i>Esta película no tiene premios. ¿Quiere saber algo más de El Caballero Oscuro La Leyenda Renace, diga si o no [...]</i> |

Fig. 44 Cartelera Puntuación (15 Agosto 2012)

En la Figura 45 se observa un diagrama sobre la gestión de la información en `billboard_search.php`.

1. En el diálogo inicial (`init.php`) el usuario solicita la información *cartelera* y cualquiera de las posibles categorías (genera, puntuación o votos).
2. La selección de la cartelera es enviada a `billboard_search.php` para su gestión. Se envía la petición a FilmAffiny y se obtiene la lista de películas de la cartelera solicitada.
3. Se informa al usuario de la cartelera y se le indica que seleccione uno de los títulos ofertados para obtener más información sobre ellos. La gestión de este diálogo también se lleva a cabo dentro del módulo `billboard_search.php`.
4. Cuando el usuario ha hecho su selección, se envían los datos al módulo de enlace (`link.php`) para que este informe al módulo de búsqueda de información, del nuevo título (`movie_search.php`).
5. Desde aquí comienza un nuevo diálogo donde el usuario pide información concreta sobre la película seleccionada.

4 DISEÑO DE LA APLICACIÓN: MovieBox

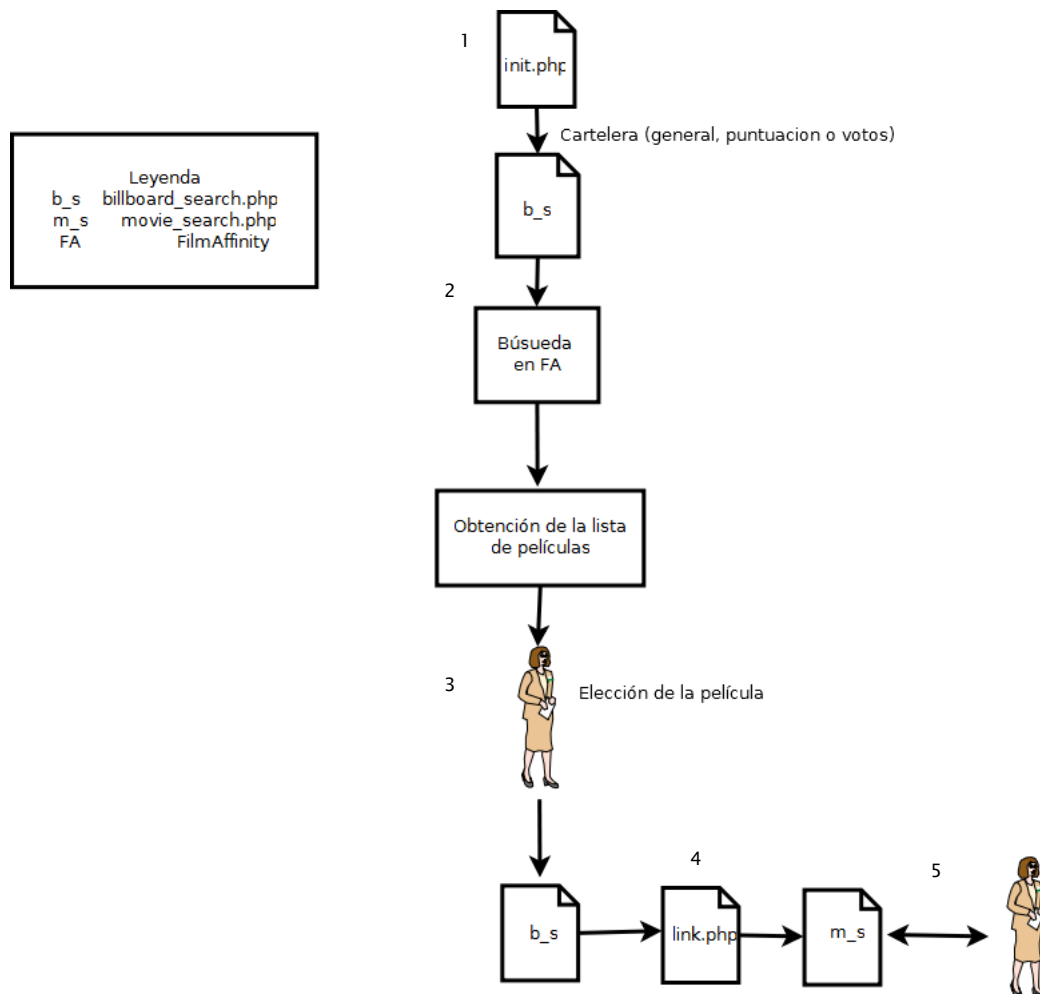


Fig. 45 Diagrama de bloques `billboard_search`

prize_search.php

En este módulo sabremos qué películas han ganado la última edición de algunas de las entregas de premios más importantes. Entre las categorías se pueden elegir festivales como el de San Sebastián, Toronto, Berlín, Moscú o Venecia. Las entregas de premios disponibles son los Oscar, los Goya y los Globos de Oro.

Acto seguido la aplicación preguntará por la información que desee saber sobre la película ganadora, al igual que ocurre al seleccionar una película de la cartelera o elegirla por título. Sigue un esquema similar al mostrado en la Figura 45, en lugar de `billboard_search.php`, se hace uso del módulo `prize_search.php`

4.3.3 Módulo auxiliar

functions.php

Este archivo no presenta ninguna interacción con el usuario directamente. En él se declaran y se ejecutan todas las funciones que son comunes al resto de módulos, tales como búsquedas, creación de gramáticas, preparación de diálogos, etc.

- **Algoritmos de búsqueda**

Dado que en esta aplicación se carece de base de datos, la información se selecciona con algoritmos de búsqueda que sean eficientes. Existen dos tipos de búsquedas: aquella cuyo fin es encontrar textos o palabras concretas y la que busca recopilar una lista de películas. Ambas están basadas principalmente en el tratamiento de cadenas de texto.

- o `search($web, $find, $iniChar, $endChar)`

Esta función devuelve el texto hallado dentro de `$web` que se encuentre a partir de la palabra clave `$find` cuyos caracteres iniciales y finales sean `$iniChar` y `$endChar`. En la Figura 46 se detalla un ejemplo donde se muestran los valores de cada una de las variables, así como el resultado final en la Figura 47.

```
$synopsis=search($web,'SINOPSIS</th>','<td>','(FILMAFFINITY)')

$web = [...] </tr> <tr> <th>SINOPSIS</th> <td>Durante una
        noche de Navidad, una anciana le cuenta a su nieta la
        historia de Eduardo Manostijeras (Johnny Depp), un
        muchacho creado por un extravagante inventor (Vincent
        Price) que no pudo acabar su obra, dejando al joven
        con cuchillas en lugar de dedos. (FILMAFFINITY)</td>
</tr> <tr> <th>CR&Iacute;TICAS</th> <td> [...]
```

Fig. 46 Llamada método de búsqueda

Esta llamada al método dará como resultado:

```
$synopsis = Durante una noche de Navidad, una anciana le
            cuenta a su nieta la historia de Eduardo
            Manostijeras (Johnny Depp), un muchacho creado por
            un extravagante inventor (Vincent Price) que no
            pudo acabar su obra, dejando al joven con
            cuchillas en lugar de dedos.
```

Fig. 47 Sinopsis resultado método de búsqueda

- o `variousResults($web, $key)`

Esta función se ejecuta cuando se busca obtener una lista de películas. En concreto, al realizar una búsqueda por título y se obtienen varias coincidencias este método almacena los títulos en un array⁶ y los números de identificación de las películas en otro para formar una matriz como la mostrada en la Tabla 6.

| Título de la película | Id |
|---|--------|
| Los juegos del hambre: Sinsajo. Parte 2 | 272141 |
| Los juegos del hambre: Sinsajo. Parte 1 | 520253 |
| Los juegos del hambre: En Llamas | 702482 |
| Los juegos del hambre | 520063 |

Tabla 6 Número de identificación de película

De forma similar sucede cuando se desea obtener la lista de películas en la cartelera.

- **Creación de gramática**

Estas gramáticas se aplican únicamente a las listas de películas. Al tratarse de una aplicación cuyos contenidos varían

⁶ Vector de almacenamiento que contiene elementos del mismo tipo. En otros lenguajes de programación puede obtener elementos de distinta naturaleza.

frecuentemente, resulta muy poco práctico disponer de gramáticas estáticas. Recordemos que las gramáticas sirven para que el reconocedor automático del habla (ASR) identifique qué es lo que el usuario está diciendo. Para evitar la continua actualización manual de estos diccionarios se ha creado la función `createDtmfGrammar($array,$field)`, donde `$array` es la lista de películas para la que se crea la gramática y `$field` es el nombre del campo dentro del cual queremos crear la gramática.

Esta función crea un fichero de extensión XML que contiene la estructura de la gramática para la selección de la película. En la Figura 48 se aprecia el código del fichero generado para la consulta de título *Gladiator*. Esta gramática se ha obtenido porque el resultado de la búsqueda es múltiple, existen 4 coincidencias de título.

```

1 <?xml versión = "1.0"?>
2 <grammar mode="voice" xml:lang="es-es" root = "peliculas"
3   xmlns="http://www.w3.org/2001/06/grammar">
4   <rule id="peliculas" scope = "public">
5     <one-of>
6       <item> 1 <tag> out.titulo="coliseo: ruedo mortal de
7         roma aka coliseo: historia de un gladiador" </tag>
8       </item>
9       <item> 2 <tag> out.titulo="gladiator el gladiador"
10      </tag> </item>
11      <item> 3 <tag> out.titulo="gladiator" </tag> </item>
12      <item> 4 <tag> out.titulo="el gladiador tv" </tag>
13      </item>
14    </one-of>
15  </rule>
16 </grammar>

```

Fig. 48 Fichero `sellItem.xml` generado para consulta 'Gladiator'

Además, en esta función se prepara la locución con la lista de películas que el TTS comunicará al usuario (Figura 49).

TSS: *Para Coliseo: ruedo mortal de Roma aka coliseo: historia de un gladiador, diga 1. Para Gladiator el Gladiador, diga 2. Para Gladiator, diga 3. Para El Gladiador, (serie de televisión), diga 4.*

Fig. 49 Diálogo resultado múltiple de la búsqueda de título

- **Modificación y formato de cadenas de caracteres**

La información obtenida debe formatearse por dos razones. La primera de ellas es debida a algunos caracteres ilegibles que provocan fallos a la hora de interpretar los textos; `normalize($string)`, donde `$string` es el texto a normalizar. La segunda es cuestión de idiomas, para evitar tener que reducir la aplicación a películas de habla hispana se implementó la función `plotLangFormat($SP, $EN,$country)`.

- o `normalize($string)`

Esta función, aunque posteriormente modificada ha sido utilizada del siguiente enlace:

<http://www.e-capty.com/reemplazar-la-n-acentos-espacios-y-caracteres-especiales-con-php>

Las tildes, tanto españolas como en cualquier idioma, algunos signos de puntuación, los paréntesis, etc. Todos ellos son eliminados del texto a interpretar por el TTS (Figura 50).

```
("\\", " ", "°", "-", "~", "#", "@", "|", "!", ";", "\"", ".", "$", "%", "&", "/", "(", ") ", "?", "¿", "¡", "í", "¿", "[", "^", "`", "]", "+", "}", "{", " ", "´", ">", "<", "quot")
```

Fig. 50 Caracteres reemplazables

- o `plotLangFormat($SP, $EN,$country)`
 - `$country` es la nacionalidad de la película, pudiendo distinguir entre francesa o estadounidense
 - `$SP` es un texto en español, normalmente será la sinopsis.

4 DISEÑO DE LA APLICACIÓN: MovieBox

- \$EN es el mismo texto pero en inglés, es decir, en el caso de la sinopsis, esta se ha recogido de la página en inglés de FilmAffinity.

La función de este método es la correcta pronunciación de los nombres de algunos de los personajes, actores y directores durante el enunciado de la información.

La principal motivación para el desarrollo de esta función fue evitar “españolizar” los nombres extranjeros. Aunque no sepamos la pronunciación exacta, sabemos reconocerlos si oímos el nombre tal y como suena en “versión original”. ¿Sabríamos que se reconocer el nombre del actor si oímos a alguien decir Michael Caine, tal y como se escribe en lugar de oír “*Maicol Quein*”?

El funcionamiento es el siguiente: si la película no es española, se comparan los textos \$SP y \$EN. El resultado de esta comparación devuelve aquellas palabras que están presentes en ambos textos. Con una alta probabilidad estas palabras serán nombres propios de personas o lugares cuya pronunciación debería ser en el idioma original. De esta manera, estas palabras quedan enmarcadas entre etiquetas de lenguaje donde se especifica en qué lengua deben ser leídos.

Figura 51: código <prompt> pronunciado exclusivamente en español. Fíjese en cómo sería la pronunciación de las palabras en negrita leídos literalmente según están escritos. Por ejemplo: Luke Skywalker como *Luque Squialquer*.

Figura 52: sinopsis de la misma película recuperada de la página de FilmAffinty en inglés.

Figura 53: resultado de la comparación de ambos resúmenes (español e inglés) incluidos dentro de la etiqueta <prompt> con sus correspondientes atributos de idioma.

```
<prompt xml:lang = "es-es">7La princesa Leia, líder del movimiento rebelde que desea reinstaurar la República en la galaxia en los tiempos ominosos del Imperio, es capturada por las Fuerzas Imperiales, capitaneadas por el implacable Darth Vader, el sirviente más fiel del Emperador. El intrépido y joven Luke Skywalker, ayudado por Han Solo, capitán de la nave espacial "El Halcón Milenario", y los androides, R2D2 y C3PO, serán los encargados de luchar contra el enemigo y e intentar rescatar a la princesa para volver a instaurar la justicia en el seno de la galaxia. </prompt>
```

Fig. 51 \$SP: sinopsis en español

In a distant galaxy, a long time ago, young Luke Skywalker, a young farmboy on Tatooine, is thrust into the struggle of the Rebel Alliance when he meets Obi-Wan Kenobi, who has lived for years in seclusion on the desert planet. Obi-Wan begins **Luke's** Jedi training as **Luke** joins him on a daring mission to rescue the beautiful Rebel leader Princess **Leia** from the clutches of the evil Empire. Although Obi-Wan sacrifices himself in a lightsabre duel with **Darth Vader**, his former apprentice, Luke proves that the Force is with him by destroying the Empire's dreaded Death Star

Fig. 52 \$EN: sinopsis en inglés

```
<prompt xml:lang = "es-es">La princesa <prompt xml:lang = "en-US">Leia</prompt>, líder del movimiento rebelde que desea reinstaurar la República en la galaxia en los tiempos ominosos del Imperio, es capturada por las Fuerzas Imperiales, capitaneadas por el implacable <prompt xml:lang = "en-US">Darth Vader</prompt>,, el sirviente más fiel del Emperador. El intrépido y joven <prompt xml:lang = "en-US">Luke Skywalker</prompt>, ayudado por Han Solo, capitán de la nave espacial "El Halcón Milenario", y los androides, R2D2 y C3PO, serán los encargados de luchar contra el enemigo y e intentar rescatar a la princesa para volver a instaurar la justicia en el seno de la galaxia.</prompt>
```

Fig. 53 Sinopsis bilingüe

⁷ Esta etiqueta indica el idioma en el que se tiene que reproducirse

link.php

Ya hemos visto este nombre en algunos diagramas de bloques. Se trata de un fichero enlace para devolver el control del diálogo a `movie_search.php`. Se precisa este archivo intermedio (Figura 54) porque es necesario añadir nuevos datos para procesar en `movie_search`.

En las líneas 12 y 13 se recogen, del diálogo precedente, el título y el identificador de la película, de esta forma, la aplicación sabe que se trata de una película única. En la línea 25, vía `<submit>` se envía la información recuperada al módulo `movie_search.php` donde se gestiona la búsqueda de información referente a la película

```

1  <?php
2  echo "<?xml version=\"1.0\" encoding=\"UTF-8\"?>";
3  ?>
4
5  <vxml version="2.1" xml:lang="es-es">
6  <?php
7
8  header('Content-Type: text/html; charset=utf8');
9  ?>
10 <?php
11     include 'functions.php';
12     $id_movie = $_REQUEST["id"];
13     $movieTitle = $_REQUEST["tit"];
14     ?>
15 <var name = "id_movie" expr="<?=$id_movie?>"/>
16 <var name = "titulo" expr="'<?=$movieTitle?>'"/>
17 <form id="linking">
18   <field name="dummy0">
19     <property name="timeout" value="1s"/>
20     <grammar type="text/gsl">[poppaoomowmow]</grammar>
21     <filled>
22       <prompt>no way this will ever happen.</prompt>
23     </filled>
24     <noinput>
25       <submit      next="movie_search.php"      method      ="get"
26         namelist="id_movie titulo" fetchaudio="GL_honor.wav"/>
27     </noinput>
28   </field>
29 </form>
30 </vxml>

```

Fig. 54 link.php

4.4 Evaluación

Con el fin de evaluar la aplicación desarrollada, se han especificado 10 preguntas a responder por 10 usuarios. Dichas preguntas miden el grado de satisfacción de los usuarios al interactuar con el sistema. Los resultados se exponen en las Tablas 7 a 16.

1. De 1 al 5, ¿Cómo calificaría sus conocimientos previos sobre las tecnologías utilizadas en la aplicación?

| | | | | | | | | | | |
|------------|-----|---|---|---|---|---|---|---|---|---|
| 1 | X | X | X | | | X | | | | X |
| 2 | | | | | X | | X | X | X | |
| 3 | | | | X | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |
| Resultado: | 1'5 | | | | | | | | | |

Tabla 7 Valoración conocimientos previos

La mayoría de los usuarios no dispone de conocimientos previos sobre las tecnologías con las que se ha desarrollado la aplicación

2. ¿Cuántas veces ha utilizado interfaces por voz previamente?

| | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|---|
| Nunca | | X | X | | | | | | | |
| Muy Pocas | X | | | | | | | | | |
| Algunas veces | | | | X | X | X | X | | | X |
| De vez en cuando | | | | | | | | X | X | |
| Muchas veces | | | | | | | | | | |

Tabla 8 Valoración uso previo de interfaces orales

Según el resultado de la encuesta, casi todos los usuarios han utilizado alguna vez una interfaz por voz.

3. ¿Qué tal entendió los mensajes generados por el sistema?

| | | | | | | | | | | |
|----------|---|---|---|---|---|---|---|---|---|---|
| Muy mal | | | | | | | | | | |
| Mal | | | | | | | | | | |
| Regular | | | | | | X | | X | | |
| Bien | X | | | X | X | | X | | X | X |
| Muy bien | | X | X | | | | | | | |

Tabla 9 Valoración comprensión de los mensajes

De acuerdo con la valoración de los usuarios, los mensajes emitidos por el TSS eran comprensibles.

4. En su opinión, la interacción con el sistema fue...

| | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|---|
| Muy lenta | | | | | | | | | | |
| Lenta | X | | | | | | | | X | |
| Adecuada | | X | X | X | X | | X | X | | X |
| Rápida | | | | | | X | | | | |
| Muy rápida | | | | | | | | | | |

Tabla 10 Valoración velocidad de interacción

La velocidad en la aplicación es adecuada según la encuesta realizada por los usuarios.

5. Califique la dificultad de utilización del sistema.

| | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|
| Muy fácil | | | | | | | | | | |
| Fácil | | X | X | X | X | | X | X | X | X |
| Regular | X | | | | | X | | | | |
| Difícil | | | | | | | | | | |
| Muy difícil | | | | | | | | | | |

Tabla 11 Valoración dificultad

A los usuarios les ha resultado, en general, fácil interactuar con la aplicación.

6. ¿Le fue fácil obtener la información que solicitaba al sistema?

| | | | | | | | | | | |
|--------------------------------|---|---|---|---|---|---|---|---|---|---|
| No, fue imposible | | | | | | | | | | |
| Sí, pero con gran dificultad | | | | | | | | | | |
| Sí, pero con alguna dificultad | | | | X | | X | | | | X |
| Sí, fue fácil | X | X | X | | X | | X | X | X | |
| Sí, fue muy fácil | | | | | | | | | | |

Tabla 12 Valoración obtención de la información

En algunas ocasiones, la obtención de la información ha sido algo dificultosa. No obstante, la tónica general devuelve buenos resultados.

7. ¿Quedó satisfecho con el uso de la aplicación?

| | | | | | | | | | | |
|----------------------------|---|---|---|---|---|---|---|---|---|---|
| Absolutamente insatisfecho | | | | | | | | | | |
| No muy satisfecho | | | | | | | | | | |
| Indiferente | | X | | | | X | | | | |
| Satisfecho | X | | X | X | X | | X | X | X | X |
| Muy satisfecho. | | | | | | | | | | |

Tabla 13 Valoración satisfacción de uso de la aplicación

La valoración general de la aplicación obtiene un resultado satisfactorio por parte de los usuarios encuestados.

8. ¿Sabía lo que tenía que responder al sistema en cada momento?

| | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|---|
| No, nunca | | | | | | | | | | |
| No, casi nunca | | | | | | | | | | |
| A veces | | | | X | | | | | X | |
| Sí, casi siempre | X | X | X | | X | X | X | X | | X |
| Sí, siempre. | | | | | | | | | | |

Tabla 14 Valoración seguimiento del diálogo

Las instrucciones dadas por el sistema se han entendido y el usuario ha sabido qué responder en la mayoría de las ocasiones.

9. ¿Cree que el comportamiento del sistema es parecido al que podría desempeñar un ser humano?

| | | | | | | | | | | |
|------------------|---|---|---|---|---|---|---|---|---|---|
| No, nunca | | | | | | | | | | |
| No, casi nunca | | | | | | | | | | |
| A veces | | | | | X | | | | | X |
| Sí, casi siempre | X | X | X | X | | X | X | X | X | |
| Sí, siempre. | | | | | | | | | | |

Tabla 15 Valoración comportammiento similar a ser humano

Los usuarios valoran el comportamiento de la aplicación parecido al de una persona, salvo excepcionales casos.

10. Valore globalmente la aplicación

| | | | | | | | | | | |
|-----------------------|---|---|---|---|---|---|---|---|---|---|
| No me ha gustado nada | | | | | | | | | | |
| Muy poco | | | | | | | | | | |
| Más o menos, buena | | X | | X | | | X | | | X |
| Realmente muy buena. | X | | X | | X | X | | X | X | |

Tabla 16 Valoración global de la aplicación

En general, la aplicación obtiene buena calificación en cuanto a la revisión global.



Capítulo 5

5 PRESUPUESTO Y PLANIFICACIÓN

5.1 Presupuesto

Seguidamente la lista de recursos empleados en el desarrollo de este Trabajo Fin de Grado.

| | |
|-------------------------|-------------------------|
| Desarrollador: | María Fernández Vallejo |
| Categoría: | Ingeniera Graduada |
| Tiempo trabajado: | 147 días (~5 meses) |
| Salario mensual: | <u>2500 €/mes</u> |
| Total personal: | 12.500 € |
| Hosting: | 0€ |
| Hardware de desarrollo: | <u>950€</u> |
| Total material: | 950 € |

El total del proyecto asciende a **13.450 €** sin IVA.

5.2 Planificación

La Tabla 17 muestra un desglose de las tareas realizadas para el diseño e implementación de este proyecto. Se incluyen las fechas de inicio y fin, así como la tarea predecesora, en caso de que exista.

| Nombre de la tarea | Inicio | Fin | Duración | Predecesora |
|---|-----------------|-----------------|-----------|-------------|
| Trabajo Fin de Grado | | | | |
| Fase previa | 10/02/12 | 27/03/12 | 33 | |
| Estudio de las tecnologías de desarrollo y herramientas | 10/02/12 | 27/03/12 | 33 | |
| Delimitación de la funcionalidad de la aplicación | 25/03/12 | 27/03/12 | 3 | |
| Fase de desarrollo preliminar | 01/04/12 | 06/04/12 | 6 | |
| Diseño estático | 01/04/12 | 03/04/12 | 3 | |
| Diseño con servidor externo | 04/04/12 | 06/04/12 | 3 | 6 |
| Desarrollo de la aplicación | 20/04/12 | 17/08/12 | 86 | |
| movie_search.php | 20/04/12 | 15/08/12 | 84 | |
| billboard_search.php | 01/06/12 | 17/08/12 | 56 | |
| prize_search.php | 10/06/12 | 13/08/12 | 47 | |
| functions.php | 14/06/12 | 06/08/12 | 38 | |
| link.php | 18/07/12 | 20/07/12 | 3 | |
| init.php | 01/06/12 | 25/06/12 | 17 | |
| Memoria | 02/08/12 | 31/08/12 | 22 | |

Tabla 17 Planificación

La Figura 55 muestra la evolución en el tiempo del desarrollo del proyecto.

5 PLANIFICACIÓN Y PRESUPUESTO

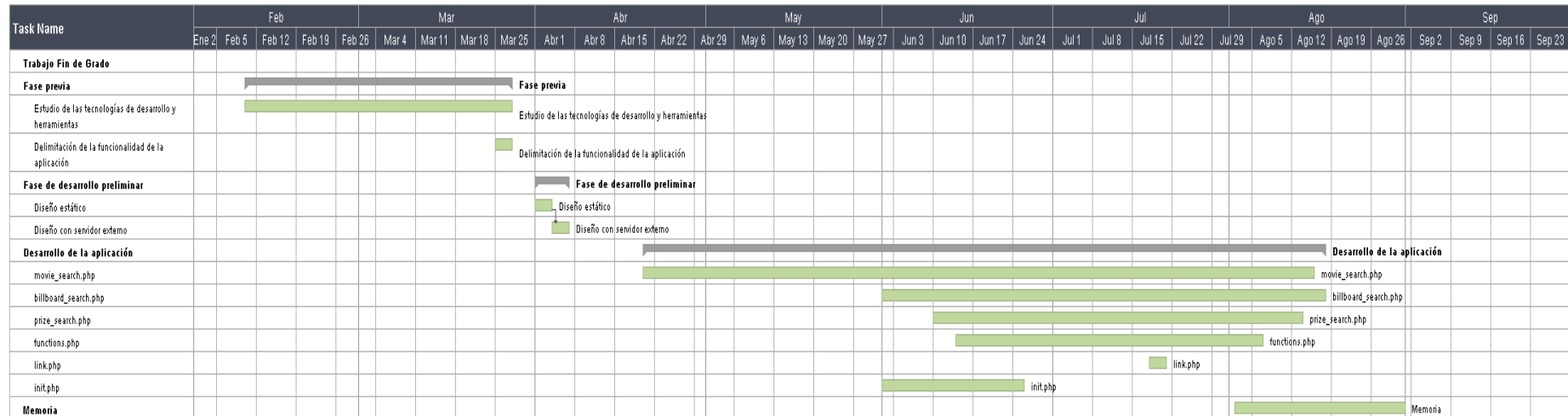


Fig. 55 Diagrama de Gantt



Capítulo 6

6 CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

Recuperemos la lista de objetivos marcados en el Capítulo 3 de este documento. Analizaremos cuántos de ellos han sido alcanzados satisfactoriamente.

- **Precisión:** aunque el sistema pueda quedar de “antipático” en este sentido, la concisión en la respuesta es máxima tal y como se lee en la Figura 56.

| | |
|----------|---|
| TSS: | <i>¿Qué quiere saber de The Guardian? Diga sinopsis, director, reparto o premios.</i> |
| Usuario: | <i>Director</i> |
| TSS: | <i>Andrew Davis. ¿Quiere saber algo más de Quiero ser italiano, diga sí o no</i> |

Fig. 56 Dialogo preciso

6 CONCLUSIONES Y TRABAJO FUTURO

Sin más sintaxis que el nombre del director. Lo que pretendía evitar es la siguiente situación, donde la frase subrayada es totalmente innecesaria (Figura 57).

| | |
|----------|---|
| TSS: | <i>¿Qué quiere saber de The Guardian? Diga sinopsis, director, reparto o premios.</i> |
| Usuario: | <i>Director</i> |
| TSS: | <i><u>El director de The Guardian es Andrew Davis.</u> ¿Quiere saber algo más de The Guardian, diga si o no</i> |

Fig. 57 Dialogo redundante

- **Brevedad en los discursos del TTS**

Exceptuando los resúmenes que puedan quedar un poco largos, sobre ese tema no podemos hacer nada; la media de longitud de los mensajes resulta adecuada para evitar la desesperación de los usuarios.

Además en caso de conocer la aplicación, les está permitido interrumpir el diálogo del TTS para agilizar los trámites de adquisición de la respuesta.

- **Instrucciones sencillas**

Dado que se trata de una aplicación informativa, las acciones a realizar por el usuario se limitan a decir un número o una palabra cuando se le pregunte.

Los mensajes que indican qué las opciones de las que dispone un usuario son presentados con una breve descripción de las mismas, en caso de que fuera necesario.

- **Claridad en las expresiones del TTS**

Como se había especificado para esta apartado, el mayor problema radicaba en las películas internacionales. Al descartarse la idea de reducir el conjunto de películas a las de habla hispana, había que pensar en un método para incluir las películas extranjeras sin que nos chirriasen los oídos al oír ciertos nombres. Esto quedó resuelto con la implementación de un método que, mediante la

comparación de dos textos, extrae los nombres comunes e indica al TTS que deben pronunciarse en otro idioma.

- **Número limitado de opciones**

De los tres menús principales, la que ofrece más opciones son los nombres de los festivales, hasta 8. No obstante, en este punto se pretendía disponer de variedad en cuanto a las posibilidades. Las películas galardonadas con un premio de estas características suelen tener un extra que anima a verlas.

Las listas de películas, se nombran hasta un máximo de 10. Era necesario encontrar el punto medio entre abarcar demasiadas opciones, o dejar sagas incompletas. El número 10 se eligió porque las películas de Harry Potter son 8 en total y no era justo que si el usuario quería saber si la primera parte tuvo premios o no, esta no se ofertara en la lista de películas. Se añadieron 2 más a favor de aumentar la información sobre las películas de la cartelera.

En definitiva, la aplicación cumple con las expectativas que se planearon. Ha resultado una aplicación muy dinámica, que era el objetivo principal, sin que por ello tengan que actualizarse manualmente los datos.

Con respecto a mi desarrollo personal, los conocimientos que he adquirido sobre los lenguajes de programación utilizados serán muy útiles a lo largo de mi vida profesional. Me llevo la satisfacción de haberlo hecho en un espacio breve de tiempo con la suficiente habilidad como para haber desarrollado esta aplicación. Además, con la redacción de este documento, he aprendido a organizar ideas y a plasmarlas en un papel de una manera ordenada, siguiendo una secuencia lógica, intentando adaptar el vocabulario más técnico a personas ajenas a la materia.

6.2 Futuras ampliaciones

La aplicación desarrollada podría considerarse como una primera aproximación de un gran sistema final. Las ideas que se exponen a continuación han sido fruto del *feedback* de algunos usuarios de la aplicación, mientras que otros han surgido durante el desarrollo de la misma.

En esta primera versión no se contempla realizar búsquedas que no sean por títulos. Es decir, no se pueden buscar nombres de directores o actores para conocer sus últimos trabajos. Esta idea podría desarrollarse para futuras actualizaciones.

A la lista de cosas por añadir, se podría investigar más acerca de reproducir audios *online*. El estándar VoiceXML únicamente puede reproducir audios que se encuentren en formato .WAV, de forma que se podría buscar un tráiler de una película en este formato y a la vez disponible en la red, y reproducirlo, obviamente sin el aporte visual, como si se escuchara por la radio.

Otra posibilidad es informar sobre los trabajos previos y futuros de los directores. Realmente, este módulo está implementado, pero finalmente no se incluyó en la versión definitiva. Esta decisión se tomó ya que el usuario medio suele saber pocos nombres de directores, y los que conoce, sabe qué películas ha hecho. Además se entraba en la disyuntiva de qué películas informar, ¿las más antiguas o las más modernas?, ¿las más premiadas tal vez? Todo esto añadía más procesado en la aplicación, y por tanto a veces podría traducirse en demasiada lentitud. No obstante, para futuras versiones podría volverse a tener en cuenta.

Al citar la cartelera, se podría acceder a los cines locales e informar sobre los horarios de los pases a las películas.

Un buen avance sería añadir un diálogo que permitiera a los usuarios registrados en FilmAffinity acceder a su cuenta. Desde ahí se

6 CONCLUSIONES Y TRABAJO FUTURO

podrían recuperar los comentarios de otros usuarios con gustos similares y recomendación de películas.

La lista podría quedar así resumida, no obstante cuantas más funcionalidades se quieran añadir, más tardará el TTS en informar al usuario sobre todas ellas.

- ⦿ Nuevos términos de búsqueda.
- ⦿ Reproducción de audio *online*.
- ⦿ Informar sobre trabajos anteriores o futuros.
- ⦿ Añadir campos de identificación de usuarios registrados en FilmAffinty.
- ⦿ Gestión de cines.

A pesar de todas las ideas, es necesario encontrar un compromiso entre dar información y la gestión del volumen de datos. Éste podría llegar a aumentar tanto que la aplicación nunca llegase a establecer comunicación con el usuario. Recordemos que los datos se recopilan antes de iniciarse el diálogo y el acceso a webs tiene sus tiempos de retardo.

Además, tal y como se ha explicado anteriormente, este tipo de aplicaciones necesitan un contexto más o menos limitado, que si bien seguimos tratando el tema cinematográfico, más posibilidades implican más potenciales respuestas de las que hay que hacerse cargo si se quiere construir un sistema robusto. Igualmente hay pensar en el perfil del destinatario final, si es una persona versada en el tema, posiblemente le interese un tipo de información distinta, más detallada que a una persona que simplemente quiere saber de qué va una película para ir o no al cine.



Anexo A

A. FAQ (Preguntas frecuentes)

Este capítulo se ha creado con la idea de solucionar pequeñas curiosidades y cuestiones que hayan podido surgir con el uso de la aplicación:

a. Le he dicho un título de una película y no me ha reconocido, me dice que no me entiende. ¿Qué pasa?

Posiblemente ese título no se encuentre dentro de la gramática. Otra posibilidad es la pronunciación incompleta del mismo. Tenga en cuenta que este tipo de sistemas son un poco estrictos con el reconocimiento del habla. El vocabulario incluido en la aplicación debe ser pronunciado tal cual para que se reconozca.

Si el título está en un idioma extranjero, la gramática estará transcrita fonéticamente en una aproximación de cómo sería la pronunciación exacta:

The Artist → *De Artist* *Glatiator* → *Gladieitor*.

En esta versión del proyecto, los nombres de película con artículo están incluidos, hay que decirlos: *Los Juegos del Hambre*, *El Rey León*...

b. ¿Por qué suena una voz distinta cuando pronuncia un nombre propio en inglés?

Esto se debe a que el sistema está diseñado para detectar, en la mayoría de las ocasiones las palabras que están en inglés o francés para articularlas con una correcta pronunciación en el idioma original. Para más información acudir a la sección **4.3.3 Módulo auxiliar: functions.php**.

c. Al elegir una película mediante un número, ¿por qué no me lleva a la anterior lista? ¿Por qué los números siempre me llevan a la lista nueva?

Cada vez que se genera una nueva lista de películas, la gramática se actualiza, creando un nuevo fichero que relaciona los números con los nuevos títulos. Para más información acudir a la sección **4.3.3 Módulo auxiliar: functions.php**

d. ¿Por qué a veces escucho más tiempo la música y otras veces tan solo el principio?

Esto se debe al estado de la red en ese momento. Al recuperar la información directamente de la web, nos exponemos a los retardos.



Bibliografía

Glosario

[BUI]

Building VoiceXML applications

<https://studio.tellme.com/vxml2/ovw/applications.html>

[Consultado en Agosto de 2012]

[CCXML]

<http://www.w3.org/TR/ccxml/>

[DON10]

Doncel, J; Olaso, J. M; Olaso, R.; Guijarrubia, V; Pérez, A; Torres, M. I. *Kiosco de información basado en un sistema de diálogo multimodal*. Procesamiento del Lenguaje Natural, Revista nº 45, septiembre 2010, pp. 311–312. Disponible en:

http://rua.ua.es/dspace/bitstream/10045/14949/1/PLN_45_35.pdf

[GLA95]

Glass, J., G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue (1995). *Multilingual spoken-language understanding in the MIT Voyager system*. Speech Communication. Revista nº 17, 1–18

[GRI08]

Griol, D., L. Hurtado, E. Segarra, and E. Sanchis (2008). *A Statistical Approach to Spoken Dialog Systems Design and Evaluation*. Speech Communication 50 (8–9), 666–682.

[HE08]

He, Y., and Young, S. (2003) *A data-driven spoken language understanding system*. Proc. of ASRU'03, pp. 583—588.

[HUR05]

Hurtado, L.-F; Blat, F; García, F; Grau, S; Griol, D; Sanchis, E; Segarra, E; Torres, F. (2005) *Sistema de diálogo para el Proyecto DIHANA*. Procesamiento del Lenguaje Natural, 35, 453–454.

Disponible en: <http://www.sepln.org/revistaSEPLN/revista/35/61.pdf>

[LOP05]

López-Cózar, R., & Araki, M. (2005). *Spoken, Multilingual and Multimodal Dialogue Systems*. John Wiley & Sons Publishers.

[MAR]

Consultado en Agosto 2012. Disponible en:

http://www.bioingenieria.edu.ar/grupos/cibernetica/milone/download/hmm_diet.pdf

[McT04]

McTear, M. F. (2004). *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer.

[PIA12]

Pieraccini, R., & Rabine, L. (2012). *The Voice in the Machine: Building Computers That Understand Speech*. MIT Press.
VoiceXML. (s.f.).

[SABLE]

<http://www.bell-labs.com/project/tts/sable.html>

[SISR]

<http://www.w3.org/TR/semantic-interpretation/>

[SRGS]

<http://www.w3.org/TR/speech-grammar/>

[SSML]

<http://www.w3.org/TR/speech-synthesis/>

[UGR]

Sistemas de diálogo hablado y multimodal. Universidad de Granada.
Disponible en http://www.ugr.es/~rlopezc/sistemas_dialogo.htm
[Consultado en Agosto de 2012]

[VOI] VoiceXML Development Guide

<http://www.vxml.org/>

[VOX]

<http://evolution.voxeo.com/>

[VOX-SKY]

Voxeo, Skype Announce New Partnership

http://www.voxeo.com/about/press_reader.jsp?date=090805

[VXML]

The Voice Browser Working Group

<http://www.w3.org/Voice/>

Manual de PHP. Disponible en: <http://php.net/manual/es/index.php>

FilmAffinity. Disponible en: <http://www.filmaffinity.com/es/main.html>

